



IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of)
MISAKA et al.)
Application Number: 10/772,438)
Filed: February 6, 2004)
For: INFORMATION PROCESSING SYSTEM AND)
OPERATING SYSTEM)
ATTORNEY DOCKET NO. HITA.0510)

Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231

LETTER

Sir:

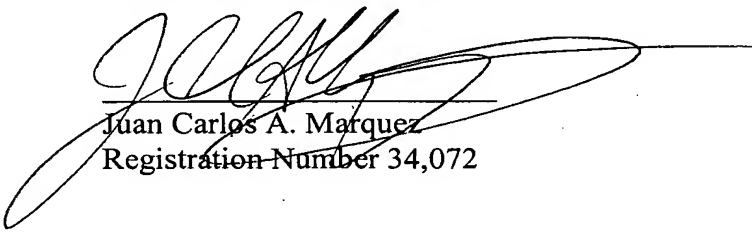
The below-identified communications are submitted in the above-captioned application or proceeding:

<input checked="" type="checkbox"/> (X)	Priority Documents (1)	
<input checked="" type="checkbox"/> (X)	Request for Priority	<input type="checkbox"/> () Assignment Document
<input type="checkbox"/> ()	Response to Missing Parts	<input type="checkbox"/> () Petition under 37 C.F.R. 1.47(a)
	w/ signed Declaration	<input type="checkbox"/> () Check for \$130.00

The Commissioner is hereby authorized to charge payment of any fees associated with this communication, including fees under 37 C.F.R. § 1.16 and 1.17 or credit any overpayment to Deposit Account Number 08-1480. A duplicate copy of this sheet is attached.

Respectfully submitted,

Stanley P. Fisher
Registration Number 24,344



Juan Carlos A. Marquez
Registration Number 34,072

REED SMITH LLP
3110 Fairview Park Drive
Suite 1400
Falls Church, Virginia 22042
(703) 641-4200
June 7, 2004



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of

MISAKA et al.

Application Number: 10/772,438

Filed: February 6, 2004

**For: INFORMATION PROCESSING SYSTEM AND
OPERATING SYSTEM**

ATTORNEY DOCKET NO. HITA.0510

**Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231**

**REQUEST FOR PRIORITY
UNDER 35 U.S.C. § 119
AND THE INTERNATIONAL CONVENTION**

Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of December 17, 2003, the filing date of the corresponding Japanese patent application 2003-418908.

A certified copy of Japanese patent application 2003-418908 is being submitted herewith. Acknowledgment of receipt of the certified copy is respectfully requested in due course.

Respectfully submitted,

Stanley P. Fisher
Registration Number 24,344

Juan Carlos A. Marquez
Registration Number 34,072

REED SMITH LLP
3110 Fairview Park Drive
Suite 1400
Falls Church, Virginia 22042
(703) 641-4200
June 7, 2004

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 1 2 月 1 7 日
Date of Application:

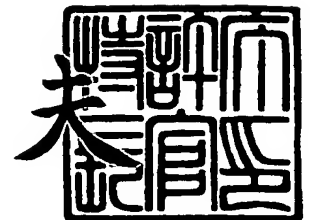
出 願 番 号 特 願 2 0 0 3 - 4 1 8 9 0 8
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 4 1 8 9 0 8]

出 願 人 株式会社日立製作所
Applicant(s):

2 0 0 4 年 2 月 6 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願
【整理番号】 H03010971A
【あて先】 特許庁長官 殿
【国際特許分類】 G06F 1/32
【発明者】
 【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所
 中央研究所内
 【氏名】 三坂 智
【発明者】
 【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所
 中央研究所内
 【氏名】 入江 直彦
【特許出願人】
 【識別番号】 000005108
 【氏名又は名称】 株式会社日立製作所
【代理人】
 【識別番号】 100075096
 【弁理士】
 【氏名又は名称】 作田 康夫
 【電話番号】 03-3212-1111
【選任した代理人】
 【識別番号】 100100310
 【弁理士】
 【氏名又は名称】 井上 学
【手数料の表示】
 【予納台帳番号】 013088
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1

【書類名】 特許請求の範囲**【請求項 1】**

プロセッサと、前記プロセッサに内蔵され又は前記プロセッサの外部に設けられるハードウェアタイマーと、前記プロセッサ及びハードウェアタイマーにクロックを供給するクロックパルス発生器とを有する情報処理装置であって、

前記ハードウェアタイマーは、でアプリケーションタスクの開始から現時点までの処理時間とを計測し、

前記アプリケーションタスクは、複数の処理単位に分割され、

前記プロセッサは、前記アプリケーションタスクの前記複数の処理単位のうち所定の処理単位を処理するのにかかった時間と前記処理単位の最悪実行時間とを比較し、前記比較結果に基づいて、前記クロックパルス発生器から出力される動作周波数を変更し、

前記プロセッサは、前記アプリケーションタスクの処理中の割込みによって起動した他のアプリケーションタスクが前記プロセッサの計算資源を占有した際に、実行準備状態のアプリケーションタスク数をカウントし、前記実行すべき他のアプリケーションタスク以外に前記実行準備状態のアプリケーションタスクが存在する場合は、前記実行準備状態監視タスクにより、前記動作周波数と前記電源電圧を上げ、存在しない場合は、前記実行準備状態監視タスクにより前記動作周波数と前記電源電圧を下げることを特徴とする情報処理装置。

【請求項 2】

請求項 1 において、

前記プロセッサは、前記アプリケーションタスクを実行する際に、前記アプリケーションタスク以外に前記実行準備状態の他のアプリケーションタスクが存在すると知らされた場合、前記アプリケーションタスクは、前記最悪実行時間よりも時刻を早めた仮想最悪実行時間と前記所定の処理単位を処理するにかかった時間とを比較することを特徴とする情報処理装置。

【請求項 3】

請求項 2 において、

前記プロセッサは、前記最悪実行時間又は前記仮想最悪実行時間と前記所定の処理単位を処理するにかかった時間を比較した結果、前記最悪実行時間又は前記仮想最悪実行時間と前記所定の処理単位を処理するにかかった時間の差が所定時間より大きい場合に、現在実行処理中の前記アプリケーションタスクを実行するための前記動作周波数と前記電源電圧を下げ、前記時間の差が所定時間より小さい場合は、前記動作周波数と前記電源電圧を上げることを特徴とする情報処理装置。

【請求項 4】

請求項 1 において、

前記プロセッサは、前記実行準備状態監視タスクへ問い合わせる事により、前記他のアプリケーションタスク以外に前記実行準備状態のアプリケーションタスクが存在すると分かた場合、前記他のアプリケーションタスクは前記最悪実行時間よりも時刻を早めた予め設定の仮想最悪実行時間、存在していないとわかった場合、元の最悪実行時間に基づいて、時間余裕の度合いを計測することを特徴とする情報処理装置。

【請求項 5】

請求項 4 において、

前記プロセッサは、前記最悪実行時間又は前記仮想最悪実行時間に基づいて計測した時間余裕の度合いが予め定められた時間余裕より大きい場合は、現在実行処理中の前記アプリケーションタスクを実行するための前記動作周波数と前記電源電圧を下げ、小さい場合は、前記動作周波数と前記電源電圧を上げることを特徴とする情報処理装置。

【請求項 6】

請求項 2 において、

前記プロセッサは、前記実行準備状態のアプリケーションタスクの数によって、前記仮想最悪実行時間における前記最悪実行時間を早める度合いを変更することを特徴とする情

報処理装置。

【請求項 7】

請求項 4 において、

前記プロセッサは、前記他のアプリケーションタスクの問い合わせに対し、前記実行準備状態のアプリケーションタスク数をカウントし、前記実行準備状態の前記アプリケーションタスクの数または有無をシステムコールで教示することを特徴とする情報処理装置。

【請求項 8】

請求項 4 において、

前記プロセッサは、オペレーティングシステムに基づいて動作することを特徴とする情報処理装置。

【請求項 9】

プロセッサと、タイマと、クロック発生回路と、動作電圧供給回路とを具備する情報処理装置であって、

前記タイマは、前記プロセッサにより実行されるアプリケーションタスクの実行時間を計測し、

前記アプリケーションタスクは、複数の処理単位に分割され、

前記情報処理装置は、実行準備状態にあるアプリケーションタスクの数を管理し、

前記情報処理装置は、前記第 1 のアプリケーションタスクを前記プロセッサを用いて実行する際に、実行準備状態にあるアプリケーションタスクが所定数より小さい場合は、前記複数の処理単位ごとに、対応する処理単位の処理が終了すべき最悪実行時間を管理し、前記第 1 のアプリケーションタスクを前記プロセッサを用いて実行する際に、実行準備状態にあるアプリケーションタスクの数が所定数より大きい場合は、前記複数の処理単位ごとに、対応する前記最悪実行時間より短い仮想最悪実行時間を管理することを特徴とする情報処理装置。

【請求項 10】

請求項 9 において、

前記情報処理装置は、第 1 の電力モードと前記第 1 の電力モードより消費電力の小さい第 2 の電力モードとを有し、前記第 1 のアプリケーションタスクの所定の処理単位が終了した時点で、前記タイマにより計測された前記所定の処理単位の実行時間と前記所定の処理単位に対応する前記最悪実行時間又は前記仮想最悪実行時間とを比較し、

前記情報処理装置は、前記所定の処理単位に対応する前記最悪実行時間又は前記仮想最悪実行時間と前記所定の処理単位の実行時間との差が所定時間より大きい場合、前記第 2 の電力モードで動作し、前記所定時間より小さい場合、前記第 1 の電力モードで動作することを特徴とする情報処理装置。

【請求項 11】

請求項 10 において、

前記プロセッサの動作周波数は、前記第 1 の電力モードより前記第 2 の電力モードの方が小さく、

前記プロセッサの動作電圧は、前記第 1 の電力モードより前記第 2 の電力モードの方が小さいことを特徴とする情報処理装置。

【請求項 12】

請求項 11 において、

前記情報処理装置は、前記第 2 の電力モードで動作している状態で、前記第 1 のアプリケーションタスクの所定の処理単位が終了した際に、実行準備状態にあるアプリケーションタスクの数が所定数より大きくなった場合に、前記第 1 の電力モードに遷移することを特徴とする情報処理装置。

【請求項 13】

プロセッサと、タイマとを有する情報処理装置の動作を制御するオペレーティングシステムであって、

前記オペレーティングシステムは、

前記プロセッサにより実行されるアプリケーションタスクを複数の処理単位に分割し、実行準備状態にあるアプリケーションタスクの数を管理し、

第1のアプリケーションタスクを前記プロセッサを用いて実行する際に、実行準備状態にあるアプリケーションタスクが所定数より小さい場合は、前記複数の処理単位ごとに、夫々の処理単位の処理が終了すべき最悪実行時間を管理し、前記第1のアプリケーションタスクを前記プロセッサを用いて実行する際に、実行準備状態にあるアプリケーションタスクの数が所定数より大きい場合は、前記複数の処理単位ごとに、対応する前記最悪実行時間より短い仮想最悪実行時間を管理するオペレーティングシステム。

【請求項14】

請求項13において、

前記情報処理装置は、第1の電力モードと前記第1の電力モードより消費電力の小さい第2の電力モードとを有し、

前記オペレーティングシステムは、前記第1のアプリケーションタスクの所定の処理単位が終了した時点で、前記タイマにより計測された前記所定の処理単位の実行時間と前記所定の処理単位に対応する前記最悪実行時間又は前記仮想最悪実行時間とを比較し、前記所定の処理単位に対応する前記最悪実行時間又は前記仮想最悪実行時間と前記所定の処理単位の実行時間との差が所定時間より大きい場合、前記情報処理装置を前記第2の電力モードで動作させ、前記所定時間より小さい場合、前記情報処理装置を前記第1の電力モードで動作することを特徴とするオペレーティングシステム。

【請求項15】

請求項14において、

前記オペレーティングシステムは、前記情報処理装置が前記第2の電力モードで動作している状態で、前記第1のアプリケーションタスクの所定の処理単位が終了した際に、実行準備状態にあるアプリケーションタスクの数が所定数より大きくなった場合に、前記情報処理装置を前記第1の電力モードに遷移させることを特徴とするオペレーティングシステム。

【請求項16】

請求項13において、

前記オペレーティングシステムは、実行準備状態にあるアプリケーションタスクの数を管理する実行準備状態監視タスクを起動し、前記実行準備状態監視タスクを前記複数の処理単位ごとに前記プロセッサにより実行することにより実行準備状態にあるアプリケーションタスクの数を管理することを特徴とするオペレーティングシステム。

【請求項17】

プロセッサと、タイマとを有する情報処理装置の動作を制御するオペレーティングシステムであって、

前記実行準備状態のアプリケーションタスクの数を出力する為のシステムコールを具備し、

前記実行すべき他のアプリケーションタスク以外に前記実行準備状態のアプリケーションタスクが存在する場合は、前記動作周波数と前記電源電圧を上げ、存在しない場合は、前記実行準備状態監視タスクにより前記動作周波数と前記電源電圧を下げることを特徴とする情報処理装置。

【書類名】 明細書

【発明の名称】 情報処理装置及びオペレーティングシステム

【技術分野】

【0001】

情報処理装置及び前記情報処理装置に搭載のソフトウェアに関するものである。

【背景技術】

【0002】

図18は、特許文献1の図5に記載の従来の電力制御装置及びその電力制御方法並びにそのプログラムを示す構成図である。前記電力制御装置は、バッテリー駆動且つプロセッサ搭載の携帯機器、例えば携帯電話、携帯型オーディオ、携帯型ビデオカメラ、ノートブックパーソナルコンピュータ、PDA (Personal Digital Assistants) の消費電力の最適化を実現する技術である。

【0003】

図18のように従来の電力制御装置には、ソフトウェア(25)で動作する中央処理装置CPU(以下、プロセッサという)(21)及び電力制御回路(22)が搭載されている。電力制御回路(22)は、電源電圧/閾値電圧制御回路、クロック周波数発生器、タイマで構成されている。前記電源電圧/閾値電圧制御回路(以下、可変DC/DC変換回路という)は、プロセッサ(21)に対して印加する電源電圧または閾値電圧を制御する機能を持っている。

【0004】

前記クロック周波数発生器(以下、CPG: Clock Pulse Generatorという)は、プロセッサ(21)に提供するクロックパルスを発生し、前記クロックパルスの時間幅、つまり動作周波数を制御する機能を持っている。タイマは、ソフトウェア(25)に対して一定時間間隔のカウント数、つまり時刻情報を提供する。前記タイマは、前記CPGからプロセッサ(21)に提供の前記クロックパルスとは別に、タイマ用クロックパルスを前記CPGから受け取り、予めソフトウェア(25)で設定の前記一定時間間隔でカウント動作を行っている。

【0005】

ソフトウェア(25)は、オペレーティングシステム(以下、RTOSという)(23)と単数ないし複数のタスク(24)等で構成されている。RTOS(23)は、単数ないし複数のタスク(24)の中からタスク起動時刻の順番に、またはタスク優先度の高い順番に、実行可能な実行準備状態のタスク(24)を保持する。更にRTOS(23)は、最初に起動されるタスク(24)の前記タスク起動時刻を現在実行中のタスク(24)に提供する機能を有する。単数ないし複数のタスク(24)は、例えばアプリケーションプログラムやミドルウェアプログラム等に相当するユーザープログラムの他に、電力制御プログラムも合わせて構成されている。

【0006】

前記電力制御プログラムは、実行中のタスク(24)に対してプロセッサ(21)に割り付けた起動時刻から現時刻までの実行時間を積算し、その積算実行時間と、予め設定した前記アプリケーションプログラムまたは前記ミドルウェアプログラムのアプリケーションスライスの最悪実行時間(以下、WCET: Worst Case Execution Timeという)から、実行中のタスク(24)における時間余裕を算出する機能を有する。アプリケーションスライスとは、タスクをN個のスライス1, 2, 3...Nに分割することをいい、スライス1, 2, 3, ..., Nの各々のWCETと、各スライスからスライスNまでのWCETは、解析や測定を通じて取得される。時間余裕があれば図18の実行中のタスク(24)からRTOS(23)に対し、所望の電源電圧、動作周波数を設定する為のシステムコールSCを発行している。そして、RTOS(23)の内部処理を行った後、ソフトウェア(25)でプロセッサ(21)から前記可変DC/DC変換回路に対して図2のコマンドCを送信する。前記コマンドCを受け取った前記可変DC/DC回路は、ソフトウェア(25)所望の電源電圧、動作周波数を設定し、制御信号CSを前記DC/DC変換回路からプロセッサ(21)に送信する。前

記制御信号CSを受け取ったプロセッサ(21)は、所望の前記電源電圧及び前記動作周波数で駆動される。

【0007】

次に、RTOS(23)上でのタスク(24)の動作状態について言及する。図19は、RTOSのタスク状態遷移図である。タスク(24)の状態には実行状態(Run状態)(31)、待機状態(Wait状態)(32)、実行準備状態(Ready状態)(33)、初期状態(34)がある。初期状態(34)は、あるタスク(24)生成以前の状態であり、既に生成のタスク(24a)からRTOS(23)提供のタスク生成用システムコールを発行して初めて、タスク(24c)が生成されて実行準備状態(33)へ遷移する。実行状態(31)はあるタスク(24)がプロセッサ(21)を占有した処理実行中の状態である。待機状態(32)は、所定の処理を終えたタスク(24d)がRTOS(23)提供のイベント受信待ち用システムコールを発行して実行状態(31)から遷移し、他のタスク(24)にプロセッサ(21)の占有を譲ってイベント待ちになる状態である。待機状態(32)のタスク(24d)は自分への起動用イベントを受信すると実行準備状態(33)に遷移する。

【0008】

実行準備状態(33)は、実行可能のタスク(24a)、タスク(24b)、タスク(24c)がタスク優先度の高い順番にプロセッサ(21)を占有する機会を伺っている状態である。そして、実行準備状態(33)のタスク(24a)、タスク(24b)、(24c)の内、その時点で最も前記タスク優先度が高いタスク(24a)が実行状態(31)へディスパッチ(35)される。また、例えば前記起動用イベントの受信によって待機状態(32)から実行準備状態(33)に遷移したタスク(24d)の前記タスク優先度が、現在実行中のタスク(24a)の前記タスク優先度よりも優先順位が高い場合、現在実行中のタスク(24a)は、前記遷移したタスク(24d)にCPUの占有を明け渡し、実行準備状態(33)にプリエンプト(36)され、同時に前記タスク優先度順に実行状態(31)になる機会を伺う。

【0009】

【特許文献1】特開2003-140787号公報

【発明の開示】

【発明が解決しようとする課題】

【0010】

本願発明者等は本願に先立って、携帯電話、PDA、携帯デジタルカメラ、携帯デジタルビデオプレイヤー、携帯デジタルビデオカメラ、携帯デジタルオーディオレコーダー及びプレイヤー等に代表される、バッテリー駆動の情報処理装置を対象に、特許文献1の前記電力制御装置の従来技術について検討した。

【0011】

図20は、本願発明者等が特許文献1に記載される電力制御装置を使用するに当たって作成した電力制御方法の概念図である。なお、図20は、特許文献1には記載されていない。縦軸は、アプリケーションプログラムにおけるアプリケーションスライスの処理経過、横軸は、アプリケーションプログラムの処理経過時刻を表している。ここで、アプリケーションスライス数はK個としている。出力時間(11)は、前記アプリケーションプログラムを仕様上完了させなければならない、または、前記アプリケーションプログラムの処理結果を出していなければならない、予め分かっている時刻である。

【0012】

特許文献1での電力制御は、電源電圧と動作周波数を制御するという技術である。従って、電力の最適化、つまり低消費電力化を出来るだけ実現するには、電源電圧を下げると同時に動作周波数も下げる必要がある。前記動作周波数を下げるとことはすなわち、ゆっくりとアプリケーションプログラムの処理を行うことに相当する。よりよく低消費電力化を行うには、出力時間(11)以内のギリギリの時刻まで低動作周波数でゆっくりと前記アプリケーションプログラムの処理を行いたい。そこで、本願発明者等は、タスク起

動時刻(12)から前記アプリケーションプログラムの処理を始めて出力時間(11)に間に合うように、線形計画法に基づいて各アプリケーションスライスに同時時間間隔のサブ出力時間(13)を割り当てることとした。また、各アプリケーションスライスに対し、タスク起動時刻(12)から各サブ出力時間の整数倍の時間間隔であるWCET(14)を予め割り当てることとした。例えば、前記線形計画法に基づいて設定したk番目のアプリケーションスライスのWCET(14c)は、サブ出力時間(13)にkを掛けた積算見積時間に相当する。

【0013】

そして、電力制御チェックポイント(6)では、タスク起動時刻(12)からのプロセッサ(21)の積算実行時間(15)と各アプリケーションスライス(4)のWCET(14)を比較し、逐次、前記電源電圧及び前記動作周波数の制御を行う。例えば、高電源電圧及び高動作周波数と低電源電圧及び低動作周波数という2つの電力の組合せで制御が可能な前記電力制御装置を用いるとする。タスク起動時刻(12)からアプリケーションスライス(4c)のk番目の処理終了時点(16)の積算実行時間(15)をアプリケーションスライス(4d)の(k+1)番目の処理のWCET(14d)から引いた時間差が、アプリケーションスライス(4d)の(k+1)番目の処理のWCET(14d)とアプリケーションスライス(4c)のk番目の処理のWCET(14c)の時間差、つまりk番目から(k+1)番目の間のサブ出力時間(13)に、前記低動作周波数に対する前記高動作周波数の倍率を掛けた時間値よりも大きい場合、k番目の処理終了時点(16)で電力を下げる。前記電源電圧及び前期動作周波数を下げても、(k+1)番目の電力制御チェックポイント(6)時のWCET(14d)の内に収まる予測出来るので、出力時間(11)厳守が満たせる。反対に小さい場合は前記電力を上げないと、つまり前記電源電圧及び前記動作周波数を上げないと、出力時間(11)厳守は満たせない。

【0014】

上述のような検討をした結果、前記情報処理装置に図18の前記電力制御装置を適用した場合、消費電力の最適化(省エネルギー化)効果について、2つの課題が残されていることがわかった。

【0015】

第1の課題は、実行中のタスク(24a)に対して、割込処理が発生した場合、起動される高優先度のタスク(24d)で費やした実行時間分、割り込まれた実行準備状態(33)のタスク(24a)の処理が遅延する為に出力時間(11)までの絶対時間に対する余裕時間にズレが生じ、電力制御のタイミングを逸してしまうという事象である。以下、この事象を図21で説明する。

【0016】

図21は、割り込まれたタスクの処理遅延を説明する図である。例えばk番目のアプリケーションスライス(4c)を処理している途中で高優先度のタスク(24d)に割り込まれた場合、割り込まれたタスク(24a)は実行準備状態(33)にプリエンブト(36)されて処理停止状態になる。割り込んだ高優先度のタスク(24b)の処理が終了すると割り込まれたタスク(24a)は実行状態(31)にディスパッチ(35)される。

【0017】

しかし、割り込まれたタスク(24a)が前記処理停止状態でも絶対時間は経過している為、予め割り当てていた(k+1)番目のWCET(14d)における絶対時間が割り込んだタスク(24d)の実行時間(37)分、遅延する。

【0018】

従って(k+1)番目での処理終了時点(16)は実行時間(37)を含んだ結果となり、絶対時間上、誤ったWCET(14)での電力制御チェックポイント(6)で電力制御を行ってしまう。また出力時間(11)も同様、実行時間(37)分、遅延する。従って省出力時間(11)も保証出来ず、メディア品質を劣化させてしまうこともわかった。

【0019】

第2の課題は、RTOS(23)の実行準備状態(33)に実行可能のタスク(24)が多

く存在している場合、実行可能のタスク (24) にはJitter (以下、ジッタという) が生じてしまう為に処理開始時刻及びWCETが遅延してしまい、時間余裕が無くなり、出力時間が守れなくなるという事象である。以下、この事象を図22で説明する。

【0020】

図22は、実行状態に遷移した直後のタスクの起動時刻の遅延を説明する図である。高優先度のタスク (24d) によって割り込まれて実行準備状態 (33) にプリエンプト (36) されたタスク (24a)、タスク (24a) によって生成した実行準備状態 (33) のタスク (24c) は、直ぐにでもプロセッサ (21) を占有するべく実行準備状態 (33) で待たされている。しかしタスク (24a)、タスク (24c) 等のようなタスク優先度が低いタスク (24) ほど、プロセッサ (21) を占有するまでの実行準備状態での待機時間 (38) は大きい。本来はタスク (24a)、タスク (24c) 等の各々で前記起動イベントを受信した時刻が起動時刻 (12) であるのだが、実際の起動時刻は待機時間 (38) 分遅延した遅延起動時刻 (39) から処理を開始し、遅延起動時刻 (39) から出力時間 (11) までしか処理時間がなくなるという課題が残っている。更に、WCET (14) は各アプリケーション/ミドルウェアプログラムが管理しているので、アプリケーション/ミドルウェアの開始が遅延すれば待機時間 (38) 分遅延してずれた遅延WCET (40) となり、最終の遅延WCET (40) も出力時間 (11) を越すという課題が残る。従って、前記積算見積時間に不確定要素が含み、省エネルギー効果が鈍くさせてしまうこともわかった。

【課題を解決するための手段】

【0021】

本願において開示される主な発明の要点は以下のとおりである。

プロセッサと、前記プロセッサに内蔵され又は前記プロセッサの外部に設けられるハードウェアタイマーと、前記プロセッサ及びハードウェアタイマーにクロックを供給するクロックパルス発生器とを有し、前記ハードウェアタイマーは、アプリケーションタスクの開始から現時点までの処理時間とを計測し、前記アプリケーションタスクは、複数の処理単位に分割され、前記プロセッサは、前記アプリケーションタスクの前記複数の処理単位のうち所定の処理単位を処理するのにかかった時間と前記処理単位の最悪実行時間とを比較し、前記比較結果に基づいて、前記クロックパルス発生器から出力される動作周波数を変更し、前記プロセッサは、前記アプリケーションタスクの処理中の割込みによって起動した他のアプリケーションタスクが前記プロセッサの計算資源を占有した際に、実行準備状態のアプリケーションタスク数をカウントし、前記実行すべき他のアプリケーションタスク以外に前記実行準備状態のアプリケーションタスクが存在する場合は、前記実行準備状態監視タスクにより、前記動作周波数と前記電源電圧を上げ、存在しない場合は、前記実行準備状態監視タスクにより前記動作周波数と前記電源電圧を下げるように構成する。

【0022】

更に望ましくは、プロセッサは、前記アプリケーションタスクを実行する際に、前記アプリケーションタスク以外に前記実行準備状態の他のアプリケーションタスクが存在すると知らされた場合、前記アプリケーションタスクは、前記最悪実行時間よりも時刻を早めた仮想最悪実行時間と前記所定の処理単位を処理するのにかかった時間とを比較する。

【0023】

更に望ましくは、前記プロセッサは、前記最悪実行時間又は前記仮想最悪実行時間と前記所定の処理単位を処理するのにかかった時間を比較した結果、前記最悪実行時間又は前記仮想最悪実行時間と前記所定の処理単位を処理するのにかかった時間の差が所定時間より大きい場合に、現在実行処理中の前記アプリケーションタスクを実行するための前記動作周波数と前記電源電圧を下げ、前記時間の差が所定時間より小さい場合は、前記動作周波数と前記電源電圧を上げる。

【0024】

別の表現をすれば、プロセッサと、タイマと、クロック発生回路と、動作電圧供給回路

とを具備し、前記タイマは、前記プロセッサにより実行されるアプリケーションタスクの実行時間を計測し、前記アプリケーションタスクは、複数の処理単位に分割され、情報処理装置は、実行準備状態にあるアプリケーションタスクの数を管理し、前記情報処理装置は、前記第1のアプリケーションタスクを前記プロセッサを用いて実行する際に、実行準備状態にあるアプリケーションタスクが所定数より小さい場合は、前記複数の処理単位ごとに、対応する処理単位の処理が終了すべき最悪実行時間を管理し、前記第1のアプリケーションタスクを前記プロセッサを用いて実行する際に、実行準備状態にあるアプリケーションタスクの数が所定数より大きい場合は、前記複数の処理単位ごとに、対応する前記最悪実行時間より短い仮想最悪実行時間を管理する。

【発明の効果】

【0025】

本願発明を実施することにより、情報処理装置の低消費電力化を向上させること達成することが出来る。

【発明を実施するための最良の形態】

【0026】

以下に本願発明を実現する実施例を説明する。

図1は、本発明の代表例を説明する図で、本発明のハードウェア構成の実施例を説明する図である。ここではバッテリー駆動で且つプロセッサ内蔵の情報処理装置を実施例として説明する。

【0027】

本明細書では、PLL (Phase Locked Loop) 回路を使用したクロックパルス発生器 (51) (以下、CPGという) 及びタイマTMU (52) 内蔵のプロセッサCPU (21) で説明する。CPG (51) ないしTMU (52) がプロセッサの外付け実装でも本願発明は実施可能である。CPG (51) は、プロセッサ (21) が同期動作をする為のクロックパルスプロセッサ (21) に供給する。プロセッサ (21) は、CPG (51) の設定で所望の動作周波数で動作を可能とする機能を具備する。また、プロセッサ (21) は、可変DC/DC 変換回路VC (22) に対して所望の電源電圧の供給するように制御信号 (57) を送信する機能を具備する。更に、プロセッサ (21) は、バス (53) でメモリ (54) と入力器及び出力器 (以下、I/Oという) (55) と接続されている。プロセッサ (21) は、メモリ (54) に格納のソフトウェアの命令手順で動作する。前記ソフトウェアは、RTOS (23)、実行準備状態監視プログラムモジュール (56)、電力/エネルギー制御プログラムモジュール (57)、アプリケーション/ミドルウェアプログラムモジュール (58) 等で構成する。

【0028】

可変DC/DC変換回路 (22) は、バッテリーBAT (59) の電池電圧をプロセッサ (21)、メモリ (54)、I/O (55) 等の各デバイスに要する電源電圧に変換し、電源系統 (60) を通して電力を前記各デバイスに供給する。また、可変DC/DC変換回路 (22) は、プロセッサ (21) のポート (62) からの制御信号 (61) を受信する機能、及びプロセッサ (21) に対してプロセッサ (21) 所望の電源電圧に可変制御する機能を具備する。I/O (55) は、例えば、静止画及び動画像を表示するディスプレイや撮像するカメラ、音を入力するマイクや出力するスピーカー等に相当する。

【0029】

図2は、本発明を実現する為のソフトウェア構成の実施例を説明する図である。RTOS (23) は、プロセッサCPU (21) の演算器等の計算資源及びメモリ (54) のメモリ資源を管理する。つまり、プロセッサ (21) に対する外部イベントやTMU (52) からのタイマイイベントによって起動するスレッド又はプロセス等と呼ばれる処理単位であるタスク (64) を割り当てて、且つ、タスク (64) が所望するメモリ容量を割り当てる。更に、各タスク (64) の間で同期、非同期動作を可能とするメッセージ通信機能を各タスク (64) に提供する。ここで、同期動作とは、一つのタスクが他のタスクに対して情報を伝達し、プロセッサにおける処理を他のタスクに受け渡す動作を言い、非同期動作とは

、一つのタスクが他のタスクに対して情報を伝達するが、プロセッサにおける処理を占有し、他のタスクに処理を受け渡さない動作を言う。

【0030】

実行準備状態監視プログラムモジュール(56)は、RTOS(23)において現時点で実行準備状態(33)になっているタスク(64)の数を把握する。実行準備状態(33)のタスク(64)の数、またはその存在の有無をシステムコール等で教示可能なRTOS(23)を搭載するならば、実行準備状態監視プログラムモジュール(56)の代用が可能であるが、本明細書では実行準備状態監視プログラムモジュール(56)を用いて本願発明を説明する。

【0031】

電力/エネルギー制御プログラムモジュール(57)は、プロセッサ(21)に印加する所望の電源電圧に設定するように、ポート(62)から制御信号(61)を可変DC/DC変換回路(22)へ送信する。更に、同時にCPG(51)に対し、所望の動作周波数に設定するよう命令する。例えば、電力/エネルギー制御プログラムモジュール(57)が高電力モード、低電力モードという2つモードで電力制御を実施する場合、高電力モードでは可変DC/DC変換回路(22)に高電源電圧を設定させる制御信号(61)を送信し、CPG(51)に高動作周波数を設定させる命令を発行する。反対に低電力モードでは可変DC/DC変換回路(22)に低電源電圧を設定させる制御信号(61)を送信し、CPG(51)に低動作周波数を設定させる命令を発行する。なお、CPG(51)の周波数を選択するためのレジスタを有する場合は、命令でなくレジスタに書き込む構成としても良い。

【0032】

アプリケーション/ミドルウェアプログラムモジュール(58)は、実際にプロセッサ(21)で処理させたい所望のプログラムであり、本願発明における電力制御の制御対象である。

【0033】

RTOS(23)、実行準備状態監視プログラムモジュール(56)、電力/エネルギー制御プログラムモジュール(57)、及びアプリケーション/ミドルウェアプログラムモジュール(58)にはそれぞれ、各々の機能呼び出す為のソフトウェアインターフェース、つまりプログラム関数群を設ける。

【0034】

本明細書では、RTOS(23)のプログラム関数群をシステムコール(System Call)(65)、実行準備状態監視プログラムモジュール(56)のプログラム関数群をレディ・ウォッチャー・インターフェース(Ready Watcher Interface)(以下、RWIという)(66)、電力/エネルギー制御プログラムモジュール(57)のプログラム関数群をパワー/エネルギー・コントロール・インターフェース(Power /Energy Control Interface)(以下、PECIという)(67)、及びアプリケーション/ミドルウェアプログラムモジュール(58)のプログラム関数群をアプリケーション・インターフェース(Application Interface)(以下、APIという)(68)と呼ぶ。RTOS(23)及び電力/エネルギー制御プログラムモジュール(57)は、プロセッサ(21)の状態レジスタ、制御レジスタ及び内部メモリ等を直接操作する。実行準備状態監視プログラムモジュール(56)は、システムコール(65)を通してRTOS(23)の機能を用いる。RWI(66)は、タスク(64)間で同期、非同期通信を行わせる為、メッセージ受信待ち処理、及びメッセージ送信処理を提供する。特にメッセージ受信待ち処理は、タスク(64)を待機状態(32)に遷移させる機能を提供する。

【0035】

アプリケーション/ミドルウェアプログラムモジュール(58)は、システムコール(65)とPECI(67)を通して、RTOS(23)及び電力/エネルギー制御プログラムモジュール(57)の機能を用いる。電力/エネルギー制御プログラムモジュール(57)は、アプリケーション/ミドルウェアプログラムのアプリケーションスライス直後の電力制御チェックポイント(6)で用いる。タスク(64)は、システムコール(65)、RWI

(66)、PECI(67)、API(68)を呼び出して所望の処理を実施する。

【0036】

本明細書では本願発明のソフトウェアの動作説明を分かりやすくする為、待機状態監視タスク(64a)、出力時間管理タスク(64b)、画像表示タスク(64c)、PCM出力タスク(64d)、ビデオ処理タスク(64e)、オーディオ処理タスク(64f)及びスリープタスク(64g)という7つのタスクで構成したAVプレイヤーを想定して説明する。実行準備状態監視タスク(64a)は、実行準備状態(33)のタスクの数を監視するためのタスクであり、システムの忙しさ(実行開始命令を発行されて直ぐにでも処理しなければならないタスクの数)を判定するためのタスクである。出力時間管理タスク(64b)は、画像表示タスク(64c)やPCM出力タスク(64d)を用いてディスプレイやスピーカーから画像表示、音声出力をしなければならない出力時間(デッドライン(Deadline))を管理し、その出力時間をRTOS(23)に把握させ、出力時間が来た際にRTOSから画像表示タスク(64c)またはPCM出力タスク(64d)に対して起動イベントを発信させるためのタスクである。画像表示タスク(64c)は、ディスプレイ等に画像表示するためのタスクであり、PCM出力タスク(64d)は、音声をスピーカー等から出力するためのタスクである。更に、ビデオ処理タスク(64e)は、例えばMPEG等の画像に関する圧縮データを解凍するためのタスクであり、オーディオ処理タスク(64f)は、例えばMP3等の音声に関する圧縮データを解凍するためのタスクである。スリープタスク(64g)は、システム全体が実行準備状態になった際にスリープ状態に入るためのタスクである。

【0037】

タスク優先度は、実行準備状態監視タスク(64a)、出力時間管理タスク(64b)、PCM出力タスク(64d)、画像表示タスク(64c)、オーディオ処理タスク(64f)、ビデオ処理タスク(64e)、スリープタスク(64g)の順に高優先度を割り付ける。

【0038】

また各タスク(64)間の同期、非同期通信は、メッセージ(71)を用いる。実行準備状態監視プログラムモジュール(56)は、予め定義したメモリフォーマットに従い、RTOS(23)使用のメモリ(54)領域からメッセージ(71)用メモリブロックを各タスク(64)に提供する。

【0039】

図3は、メッセージフォーマットを説明する図である。メッセージフォーマットのメッセージ内容は、受信タスクのIDに相当する送信先宛先TO-ID(72)、送信タスクのIDに相当する送信元宛先FR-ID(73)、送信内容(74)で構成する。各タスク(64)は所望の処理を行う為、前記メッセージフォーマットに従って前記メッセージ内容をメッセージ(71)に相当する前記メモリブロックに書き込む。そして、各タスク(64)はメッセージ(71)を送受信しながら所望の処理を行う。

【0040】

次に、本願発明であるマルチタスク処理に対応した電力制御方法について説明する。図4は本発明の概念図である。ここでは、簡単のため、高電力モードと低電力モードの2つがあるものとして説明する。まず、仮想出力時間(41)を設定し、仮想最悪出力時間(以下、仮想WCET)(42)を設定する。仮想出力時間(41)は、出力時間(11)より短くなるように設定する。ここでは、仮想出力時間(41)を出力時間(11)の半分の時刻として予め設定し、前記線形計画法により、仮想WCET(42)を設ける。仮想WCETは、同じアプリケーションスライス(4)に対してWCET(14)の半分の時刻、仮想サブ出力時間(43)もサブ出力時間(13)の半分の時間となる。

【0041】

RTOS(23)または実行準備状態監視タスク(64e)は、実行準備状態(33)にあるタスク(24)の数をカウントする。RTOS(23)または最高タスク優先度の実行準備状態監視タスク(64e)が実行準備状態のタスク(24)を監視した結果、実行準備状態(33)のタスク(24)がスリープタスク(64g)以外に全く無い場合は、低電源電圧、低動作周波数とする。これにより、情報処理装置の低電力化を測ることが可能とな

る。また、スリープ状態に入ることが可能な場合は、スリープタスク（64g）を実行し、スリープ状態に遷移する。

【0042】

逆に、実行準備状態（33）のタスク（24）がスリープタスク（64g）以外に1つ存在する場合は、消費電力を低減するために低電力モードとし、タスクを実行する。実行準備状態（33）にあるタスクが2つ以上存在する場合は、実行中のタスクを早く終了し、実行準備状態（33）にあるタスクに処理を早く受け渡すために高電力モードに設定する。

【0043】

実行中のタスク（24）は、各電力制御チェックポイント（6）で先ずRTOS（23）または実行準備状態監視タスク（24e）に問い合わせ、現時点で実行準備状態（33）のタスク（24）が存在するかどうかを判別する。判別した結果、実行準備状態（33）のタスク（24）が存在する場合は仮想WCET（42）と比較して時間に間に合っているか、存在しない場合はWECT（14）と比較して時間に間に合っているかで電力の制御を実施する。即ち、時間に間に合っていない場合は高電力モードに遷移し、時間に間に合っている場合は低電力モードに遷移する。処理を早く終了させ、実行準備状態（33）にあるタスク（24）に処理を受け渡すためである。このように実行準備状態（33）のタスクがある場合に仮想WCETと比較することにより、早く実行中のタスクを終了することが出来、実行準備状態のタスクにCPUを受け渡すことが可能になる。

【0044】

なお、本実施例では、WCET及び仮想WCETを出力時間に合わせて定めたが、これに制限されないことは当然である。例えば、現実の出力時間より早い時間にあわせてWCET及び仮想WCETを定めても良い。このように構成することにより、現実の出力時間までのマージンを取ることで、品質保持の信頼性を向上させることが可能となる。

【0045】

ここで、図5のアプリケーションスライスについて説明する。タスク（24）は、アプリケーションプログラム開始点（1）から初期化処理（2）を経て、アプリケーションプログラム終了点（3）までの処理を実行する。初期化処理（2）は、アプリケーションプログラム、ミドルウェアプログラム、デバイスドライバプログラム等で使用するメモリ領域の確保や初期化、更にはプロセッサ（21）で使用する電源電圧、動作周波数及び各レジスタの値の設定等、プロセッサ（21）上でアプリケーションを動かす為に必要な前処理を行っている。

【0046】

マルチメディア処理を扱うアプリケーションプログラム及びミドルウェアプログラムは同処理を繰り返し行う性質がある。繰り返し行われる前記同処理の単位をアプリケーションスライス（4）と呼ぶ。図5のアプリケーションスライス（4a）から（4e）の中の値はアプリケーションスライス（4）の繰り返し処理数を表しており、前記値はアプリケーション及びミドルウェアプログラムの内部に実装のアプリケーションスライス処理経過管理部（7）に逐次、格納している。

【0047】

また、繰り返し判別処理（5）ではアプリケーションスライス（4）の前記繰り返し処理数を判別し、全部のアプリケーションスライス処理を終えたらアプリケーションプログラム終了点（3）でアプリケーションが終了する。

【0048】

図6は、MPEG-4 ビデオ Decoderプログラムのアプリケーションスライスを説明する図である。例えばQCIFと呼ばれる縦144画素、横176画素の画像フォーマット（8）にMPEG-4ビデオ Decoderの処理を施すと、MPEG-4 ビデオ Decoderプログラムは画像フォーマット（8）をマクロブロック（9）と呼ばれる縦16画素、横16画素の画素ブロック毎に画像へ復号化を施す。従って、前記MPEG-4 ビデオ Decoderプログラムのアプリケーションスライス処理は、マクロブロック毎の画像復号化処理に相当し、アプリケーション

スライス（４）の繰り返し処理数はQCIFの画像フォーマット（８）中のマクロブロック数と同数の９９回となり、繰り返し判別処理（５）は９９回、アプリケーションスライス処理を行ったかどうかを判別する。

【0049】

画像フォーマット（８）が縦９６画素、横１４４画素であるSub QCIFの場合はマクロブロック数が４８となるので、アプリケーションスライス数は４８となる。

また、前記MP3 Decoderプログラムの場合、アプリケーションスライス（４）の前記繰り返し処理数はチャンネル数に相当する。ステレオ再生ならば前記繰り返し処理数は２回である。言い換えれば、音声音響復号化処理であるMP3 Decoderプログラムの場合、チャンネル単位のDecode処理を繰り返しがアプリケーションスライス（４）となる。

【0050】

図７は、実行準備状態監視タスク動作のフローチャートを説明する図である。RTOS（２３）提供のシステムコール（６５）で生成した実行準備状態監視タスク（６４a）は、初期状態（８１）で実行準備状態監視変数部（８２）を０に初期化した後、RWI（６６）からメッセージ受信待ち判別処理（８３）を呼び出す。

メッセージ受信待ち判別処理（８３）は、実行準備状態監視タスク（６４a）に対する任意の起動イベントであるメッセージ（７１a）が届いているかを判別し、届いていたら実行準備状態監視タスク（６４a）所望の処理である実行準備状態把握処理（８４）を行い、届いていなければメッセージ（７１a）が受信されるまで実行準備状態監視タスク（６４a）を待ち状態にする。

【0051】

図８は、実行準備状態把握処理のフローチャートを説明する図である。実行状態把握処理（８４）は、実行準備状態監視開始点（９１）から処理を開始し、実行準備状態監視タスク（６４a）への送信されたメッセージ（７１a）のメッセージ判別処理（９２）を行う。

【0052】

メッセージ判別処理（９２）では、受信したメッセージ（７１a）が、実行準備状態遷移用メッセージか、待機状態遷移用メッセージかを判別する。

前記実行準備状態遷移用メッセージは、例えば、図８のメッセージフォーマットのパラメータAの箇所に送信内容（７４a）に実行準備状態への遷移を表す１が書き込まれ、反対に前記待機状態遷移用メッセージは、待機状態への遷移を表す０が書き込まれたものと予め決めておく。

【0053】

従って、メッセージ判別処理（９２）は、受信したメッセージ（７１a）のパラメータAの送信内容（７４a）が１か０かを判別する。もし、１ならば、実行準備状態加算処理（９３）を行い、０ならば実行準備状態減算処理（９４）を行う。実行準備状態加算処理（９３）は、実行準備状態監視プログラムモジュール（５６）内の実行準備状態監視変数部（８２）に１を加算し、受信したメッセージ（７１a）の送信先宛先（７２）が記されていたら送信先宛先（７２）に再送信し、記されていなかったら受信したメッセージ（７１a）を廃棄、つまりメモリに返却する。反対に実行準備状態減算処理（９４）は、実行準備状態監視変数部（８２）に１を減算し、受信したメッセージ（７１a）を廃棄、つまりメモリに返却する。

【0054】

その後、処理は実行準備状態監視変数判別処理（９５）に進行する。もし、実行準備状態監視変数判別処理（９５）で実行準備状態監視変数部（８２）が０または１であれば、PECI（６７）から電力/エネルギー制御プログラムモジュール（５７）機能である低電力モード処理（９６）を呼び出し、それ以外であれば同じく高電力モード処理（９７）を呼び出す。

【0055】

図９は、低電力モード処理及び高電力モード処理のフローチャートを説明する図である

。低電力モード処理(96)は、低電力モード処理開始点(101)から開始し、低動作周波数変更処理(102)、低電源電圧変更処理(103)を順に処理をする。低動作周波数変更処理(102)は、プロセッサ(21)に供給する動作周波数を低く設定するよう、CPG(51)から発生のカロックパルスを変更する。低電源電圧変更処理(103)は、プロセッサ(21)に供給する電源電圧を低くする制御信号(61)をプロセッサ(21)から可変DC/DC変換回路(22)に送信する。可変DC/DC変換回路(22)は制御信号(61)を受信すると前記電源電圧を低くする。

【0056】

高電力モード処理(97)は、高電力モード処理開始点(104)から開始し、高電源電圧変更処理(105)、高動作周波数変更処理(106)を順に処理をする。高電源電圧変更処理(105)は、プロセッサ(21)に供給する電源電圧を高くする制御信号(61)をプロセッサ(21)から可変DC/DC変換回路(22)に送信する。可変DC/DC変換回路(22)は、制御信号(61)を受信すると前記電源電圧を高くする。高動作周波数変更処理(106)は、プロセッサ(21)に供給する動作周波数を高く設定するよう、CPG(51)から発生のカロックパルスを変更する。

【0057】

図10は、ビデオ処理タスク及びオーディオ処理タスク動作のフローチャートを説明する図である。RTOS(23)提供のシステムコール(65)で生成したビデオ処理タスク(64e)は、先ずビデオ処理初期化開始点(111)から動作を開始し、RWI(66)からメッセージ受信待ち判別処理(83)を呼び出す。

【0058】

メッセージ受信待ち判別処理(83)はビデオ処理タスク(64e)に対する起動イベントであるメッセージ(71e)が届いているかを判別し、届いていたらビデオ復号化処理(112e)を行い、届いていなければメッセージ(71e)が受信されるまでビデオ処理タスク(64e)を待ち状態(32)に移す。

【0059】

ビデオ復号化処理(112e)は、図4、図5、図6に示すマクロブロック処理毎のアプリケーションスライス(4)の繰り返し処理(5)と電力制御チェックポイント(6)での電力制御を行う。ビデオ復号化処理(112e)後は、RWI(66)から待機状態遷移処理(113)を呼び出す。待機状態遷移処理(113)は、ここではビデオ復号化処理(112e)が終了したことを実行準備状態監視タスク(64a)に知らせる為、前記待機状態遷移用メッセージであるメッセージ(71a)を実行準備状態監視タスク(64a)に送信し、最後にビデオ処理タスク(64e)を待機状態(32)に遷移させる。

【0060】

またRTOS(23)のシステムコール(65)で生成したオーディオ処理タスク(64c)は、ビデオ復号化処理(112e)をオーディオ復号化処理(112f)に置換したものに相当する。従って、ステレオまたはチャンネル処理毎のアプリケーションスライス(4)の繰り返し処理(5)終了後、電力制御チェックポイント(6)で電力制御を行う。

【0061】

図11は、タスク内部で行う電力制御アルゴリズムのフローチャートを説明する図である。本明細書では、ビデオ復号化処理(112e)及びオーディオ復号化処理(112f)の内部にアプリケーションスライス(4)があり、電力制御チェックポイント(6)で電力制御アルゴリズムソフトウェアを埋め込む。

【0062】

先ず、タスク内電力制御開始点(121)から処理を開始し、実行準備状態監視変数部(82)について、実行準備状態監視変数判別処理(122)を行う。実行準備状態監視変数判別処理(122)では、実行準備状態監視変数部(82)が1かどうかを判別する。もし1ならばWCET比較処理(123)、1以外ならば仮想WCET比較処理(124)を行う。WCET比較処理(123)は、例えばビデオ処理タスク(64e)ないしオーディオ処理タスク(64f)のタスク起動時刻(12)からアプリケーションスライス(4c)のk

番目の処理終了時点(16)の積算時間(15)をアプリケーションスライス(4d)の(k+1)番目の処理のWCET(14d)から引いた時間差と、前記アプリケーションスライス(4d)の(k+1)番目の処理のWCET(14d)と前記アプリケーションスライス(4c)のk番目の処理のWCET(14c)の時間差、つまりk番目から(k+1)番目の間のサブ出力時間(13)に前記低動作周波数に対する前記高動作周波数の倍率を掛けた時間値を比較する。積算時間(15)は、TMU(52)を用いてカウントして時間を積算する。

【0063】

もし、WCET(14d)から積算実行時間(15)を引いた値の方が前記倍率を掛けた時間よりも大きい場合は、電力/エネルギー制御プログラムモジュール(57)内部管理の電力制御モード部(125)を1にする。反対に小さい場合は0にする。更に、アプリケーション/ミドルウェアプログラムモジュール(58)内部管理のWCET格納部(126)の値にサブ出力時間(13)の値を加算する。なお、WCET(14)はWCET格納部(126)の値である。また、WCET格納部(126)は初期化处理(2)でサブ出力時間(13)とする。

【0064】

一方、仮想WCET比較処理(124)は、前記と同様、例えば、ビデオ処理タスク(64e)ないしオーディオ処理タスク(64f)のタスク起動時刻(12)からアプリケーションスライス(4c)のk番目の処理終了時点(16)の積算時間(15)をアプリケーションスライス(4d)の(k+1)番目の処理の仮想WCET(42d)から引いた時間差と、前記アプリケーションスライス(4d)の(k+1)番目の処理の仮想WCET(42d)と前記アプリケーションスライス(4c)のk番目の処理の仮想WCET(42c)の時間差、つまりk番目から(k+1)番目の間の仮想サブ出力時間(43)に前記低動作周波数に対する前記高動作周波数の倍率を掛けた時間値を比較する。同じく積算実行時間(15)はTMU(52)を用いてカウントして時間を積算する。

【0065】

もし、仮想WCET(43d)から積算時間(15)を引いた値の方が前記倍率を掛けた時間よりも大きい場合は、電力/エネルギー制御プログラムモジュール(57)内部管理の電力制御モード部(125)を1にする。反対に小さい場合は0にする。更にアプリケーション/ミドルウェアプログラムモジュール(58)内部管理の仮想WCET格納部(127)の値にサブ仮想出力時間(13)の値を加算する。なお、仮想WCET(42)は仮想WCET格納部(127)の値である。また仮想WCET格納部(127)は初期化处理(2)で仮想サブ出力時間(43)とする。

【0066】

電力制御モード判別処理(128)は、電力制御モード部(125)を判別する。もし電力制御モード部(125)が1ならば低電力モード処理(96)に進み、0ならば高電力モード処理(97)に進む。

図12は、出力時間管理タスク動作のフローチャートを説明する図である。出力時間管理タスク(64b)は、出力時間管理開始点(131)から処理を開始する。出力時間管理開始点(131b)では、初期化处理として出力時間管理タスク(64b)自身に再度、起動イベントとしてメッセージ(71b)を送信する。

【0067】

メッセージ受信待ち判別処理(83)は、出力時間管理タスク(64b)へ起動イベントとしてメッセージ(71b)が届いているかを判別し、届いていたら出力時間管理処理(132)を行い、届いていなければメッセージ(71b)が受信されるまで出力時間管理処理タスク(64b)を待ち状態(32)に移す。

【0068】

出力時間管理処理(132b)は、1フレームのビデオ表示時刻ないし1フレームのオーディオ出力時刻を予めRTOS(23)に認識させる。その直後、RWI(66)から待機状態遷移処理(113)を呼び出す。待機状態遷移処理(113)を呼び出した時点で出力時

間管理処理タスク (64b) は待機状態 (32) に遷移する。

前記ビデオ表示時刻または前記オーディオ出力時刻になると、RTOS (23) から起動するタイマイベントとしてメッセージ (71c) を画像表示タスク (64c) に、またはメッセージ (71d) をPCM出力タスク (64d) に送信する。

【0069】

画像表示タスク (64c) 及びPCM出力タスク (64d) のフローチャートは図12で説明の出力時間管理タスク (64b) と同じである。しかし、出力時間管理処理 (132b) の部分が、画像表示タスク (64c) では画像表示処理 (132c)、PCM出力タスク (64d) ではPCM出力処理 (132d) となる。そして、出力時間管理開始点 (131b) では初期化処理として出力時間管理タスク (64b) 自身にメッセージ (71b) を送信するが、画像表示開始点 (131c)、PCM出力開始点 (131d) ではメッセージ (71) は送信しない。

【0070】

画像表示処理 (132c) は、ビデオ処理タスク (64e) で処理した画像をディスプレイに表示する。そして実行準備状態監視タスク (64a) に対してビデオ処理タスク (64e) の送信先宛先 (72) を記した前記実行準備状態遷移用メッセージ (71e) を送信し、更に実行準備状態監視タスク (64a) に受信した送信先宛先 (72) がビデオ処理タスク (64e) である実行準備状態遷移用メッセージ (71e) をビデオ処理タスク (64e) へ送信させる。

【0071】

PCM出力処理 (132d) は、オーディオ処理タスク (64f) で処理したPCM信号をスピーカーに表示する。そして実行準備状態監視タスク (64a) に対してオーディオ処理タスク (64f) の送信先宛先 (72) を記した前記実行準備状態遷移用メッセージ (71f) を送信し、更に実行準備状態監視タスク (64a) に受信した送信先宛先 (72) がオーディオ処理タスク (64f) である実行準備状態遷移用メッセージ (71f) をオーディオ処理タスク (64f) へ送信させる。

【0072】

図13は、スリープタスク動作のフローチャートを説明する図である。スリープタスク (64g) のスリーププログラム開始点 (141) から開始し、スリープ処理 (142) に進む。スリープ処理 (142) は、プロセッサ (21) に供給のクロックパルスを停止するようにCPG (51) で設定し、プロセッサ (21) を停止させる。スリープタスク (64g) は最低優先度タスクとしていたので、スリープタスク (64g) 以外の全タスクが待機状態、つまりアイドル状態になったらプロセッサは停止する。またTMU (52) またはI/O (55) からの割込みが生じるとアイドル状態から通常の動作状態に復帰する。

【0073】

以上の実装を行うと、実行中のタスクに割込み及び例外処理が入っても、タスク処理の遅延を抑制する制御が可能であり、電力の最適化に効果がある。

図14は、本発明の情報処理装置の電力制御動作の一実施例を説明する図である。縦軸は、タスク (64) の消費電力、横軸は時刻とし、各タスクの処理経過を表している。因みにタスク毎の面積は各タスクの消費エネルギーに相当する。

【0074】

例えばビデオ処理タスク (64e) の処理は、タスク起動時刻 (12e) で開始し、ビデオ処理タスク (64e) のデッドラインである出力時間 (11e) まで行う。この時、実行準備状態監視タスク (64a) の初期化処理 (2) は、実行準備状態監視変数部 (82) を1、タスク起動時刻 (12e) での情報処理装置の電力を低電力と設定する。ビデオ処理タスク (64e) の処理終了までの過程で、アプリケーションスライス (4) の処理毎に電力制御を行う。例えば、3つ目のアプリケーションスライスの処理が終了した段階では、WCETと比較して処理が遅れているため高電力モードに遷移する。

【0075】

予め設定していたPCM出力タスク (64d) の起動時刻 (12d) になると、優先度が高

いPCM出力タスク(64d)が起動される。また、実行準備状態監視変数部(82)に1が加算されて2となる。実行準備状態監視変数部(82)が2となるので実行準備状態監視タスク(64a)は、情報処理装置を高電力モードにする。そしてビデオ処理タスク(64e)はプリエンプト(36)とされ、実行準備状態(33)となる。PCM出力タスク(64d)は、実行状態(31)となり、予め復号化処理で生成したオーディオのPCM信号をスピーカに出力する。出力を終了するとオーディオ処理タスク(64f)にメッセージ(71f)を送信することでオーディオ処理タスクを実行準備状態とし、図示しないが、実行準備状態監視変数部(82)に1が加算されて3となる。

【0076】

PCM出力タスク(64d)は、終了直前にメッセージ(71a)を実行準備状態監視タスク(64a)にメッセージを送信し、実行準備状態監視変数部(82)に1を減算して2にする。

オーディオ処理タスク(64f)の起動時刻(12f)からオーディオ処理タスク(64f)の処理が始まるが、実行準備状態監視変数部(82)が2となっているので、オーディオ処理タスク(64f)の起動時の情報処理装置の電力は高電力モードのままである。

【0077】

また、オーディオ処理タスク(64f)は、実行準備状態監視変数部(82)が2となっているので、オーディオ処理タスク(64f)の出力時間(11f)ではなく、オーディオ処理タスク(64f)の仮想出力時間(41f)で、アプリケーションスライス(4)の処理毎に電力制御を行う。オーディオ処理タスク(64f)は、終了直前にメッセージ(71a)を実行準備状態監視タスク(64a)にメッセージを送信し、実行準備状態監視変数部(82)に1を減算して1にする。

【0078】

オーディオ処理タスク(64f)が終了すると今までプリエンプト(36)されて実行準備状態であったビデオ処理タスク(64e)が実行状態(31)に復帰し、続きの処理を開始する。復帰時の実行準備状態監視変数部(82)は1であるので、実行準備状態監視タスク(64a)は情報処理装置の電力の低電力を設定する。

【0079】

ビデオ処理タスク(64e)は、実行準備状態監視変数部(82)が1であるので、ビデオ処理タスク(64e)の仮想出力時間(41e)ではなく出力時間(11e)でアプリケーションスライス(4)毎に電力制御を行う。

ビデオ処理タスク(64e)がビデオ処理タスク(64e)の出力時間(11e)で終了すると終了直前にメッセージ(71a)を実行準備状態監視タスク(64a)にメッセージを送信し、実行準備状態監視変数部(82)に1を減算して0にする。

【0080】

実行準備状態監視変数部(82)は0であるので、実行準備状態監視タスク(64a)は、情報処理装置の電力モードを低電力に設定する。

予め設定していた画像表示タスク(64c)の起動時刻(12c)になると、画像表示タスク(64c)が起動し、実行準備状態監視変数部(82)に1が加算されて1となる。実行準備状態監視変数部(82)は1なので、実行準備状態監視タスク(64a)は低電力のままにする。

【0081】

画像表示タスク(64c)は、画像をディスプレイに表示した後、メッセージ(71a)を実行準備状態監視タスク(64a)に送信し、実行準備状態監視変数部(82)に1を減算して0にする。

オーディオ処理タスク(64f)の出力時間(11f)後、予め設定していたPCM出力タスク(64d)の起動時刻(12d)になると、実行準備状態監視変数部(82)に1が加算されて1となる。

【0082】

上記の制御の効果により、電源電圧及び動作周波数の制御の切り替えタイミングを最適

にすることが可能となる。そして消費電力は電源電圧の2乗と動作周波数の積に比例することが一般的に知られている。例えば、電源電圧及び動作周波数の組合せを、1.25[V]、60[MHz]、と1.5[V]、120[MHz]の2電力モードとして電力制御を行う場合、電源電圧の比率は0.83、動作周波数の比率はとなることから、常に1.5[V]及び120[MHz]で動作の消費電力と比較し、約65%の消費電力を下げる事が可能である。また消費エネルギーは電源電圧の2乗と低電源電圧での実行サイクルの積に比例することから、低電源電圧制御と高電源電圧制御の実行サイクル比を50%から90%と仮定すると、常に1.5[V]及び120[MHz]で動作の消費エネルギーと比較し、約15%から約25%の消費エネルギーを省けることが可能である。

【0083】

図15は、本発明の第2の実施例を示す概念図である。図4と異なる点は、複数の仮想WCET(42)を設けた点にある。第1の実施例では、簡単のため、実行準備状態(33)のタスク(24)があるか否かでWCET(14)を使用するか仮想WCET(42)を使用するかを決定している。それに対し、本実施例では、仮想WCETを複数設け、実行準備状態(33)にあるタスク(24)の数が多い場合は、より短い時間内に終わらせるよう制御するものである。例えば、実行準備状態のタスクが3つ場合(R=3)は、右から3番目の仮想WCETと比較する。このように複数の仮想WCET(42)を設け、実行準備状態(33)にあるタスク(24)の数に対応して選択することにより、よりきめ細かい電力制御が可能となる。また、実行準備状態(33)にあるタスク(24)の数が多いということは、プロセッサ(21)が処理しなければならないタスク(24)が多いということに該当する。本実施例では、実行準備状態(33)にあるタスク(24)が多いほど、早く処理を終了するように電力モードを制御するため信頼性の高い電力制御システムが提供できる。

【0084】

図16は、本発明を施した1プロセッサ構成の情報処理装置の一実施例を説明する図である。特に1プロセッサ構成の携帯電話への適用を想定した。

図1のプロセッサCPU(21)に相当する部分は、ベースバンドプロセッサBB-CPU(21b)である。ベースバンドプロセッサ(21b)は主にデジタル通信処理及び通信プロトコル処理を実施するが、通信処理を行っていない時はアプリケーション処理も行う。ベースバンドプロセッサ(21b)の端子は直接、アンテナ(151)に繋がれたアナログ無線回路(以下、RFという)(152)、通話する為のマイクロフォン(153)、通話と音楽を再生する為のスピーカー(154)及びバス(53)と接続する。

図1のI/O(55)に相当する部分は、図16ではCMOSまたはCCDカメラCAM(155)と液晶または有機ELディスプレイDPY(156)に置換すればよい。

【0085】

またメモリ(54)は、フラッシュメモリに代表される不揮発メモリFLMEM(54a)とSRAMに代表される揮発メモリSRAM(54b)を用いる。

不揮発メモリ(54a)には、プログラム命令の本体と固定変数等の静的プログラムデータを格納する。つまり、RTOS(23)、実行準備状態監視プログラムモジュール(56)、電力/エネルギー制御プログラムモジュール(57)、アプリケーション/ミドルウェアプログラムモジュール(58)を搭載することになる。

【0086】

一方、揮発メモリ(54b)は、プログラム命令によって読み書きされる動的変数の実現の為に使用する。つまり揮発メモリ(54b)は、RTOS(23)の動的メモリエリア(157a)、実行準備状態監視プログラムモジュール(56)の動的メモリエリア(157b)、電力/エネルギー制御プログラムモジュール(57)の動的メモリエリア(157c)、アプリケーション/ミドルウェアプログラムモジュール(58)の動的メモリエリア(157d)を具備する。

【0087】

図17は本発明を施した2プロセッサ構成の情報処理装置の一実施例を説明する図である。図16の情報処理装置では、通信信号処理及び通信プロトコル処理とマルチメディア

処理を同時に処理する必要が起こった場合、ベースバンドプロセッサBB-CPU (21b) にかかる負荷が非常に重い。そこで、通信処理用途のベースバンドプロセッサ (21b) とマルチメディア処理用途のアプリケーションプロセッサAP-CPU (21a) を情報処理装置に搭載している。ここでもベースバンドプロセッサ (21b) とアプリケーションプロセッサ (21a) の両者に本願発明を適用することが可能である。

【0088】

以上、実施例に基づいて説明してきたが、本発明の要旨を逸脱しない限り種々の変更が可能である。例えば、電力モードを低電力モードと高電力モードの2種類で説明してきたが、これに制限されない。即ち、電力モードを複数設け、WCET又は仮想WCETと各アプリケーションスライスの処理終了時の積算実行時間との差、即ち、時間的余裕によって、複数の電力モードから一つの電力モードを選択する構成としてもよい。このように構成することで、よりきめ細かい電力制御を行うことが可能となる。また、電力モードは、モードによって動作周波数と電源電圧とを変更したが、例えば、電源電圧は変えずに動作周波数のみを変更しても良い。

【産業上の利用可能性】

【0089】

本発明は、情報処理装置、特に、情報処理通信装置に適用して有用なものである。

【図面の簡単な説明】

【0090】

【図1】 本発明の代表例を説明する図である。

【図2】 本発明を実現する為のソフトウェア構成の実施例を説明する図である。

【図3】 メッセージフォーマットを説明する図である。

【図4】 本発明の概念図である。

【図5】 アプリケーションスライスの概念図である。

【図6】 MPEG-4 ビデオ Decoderプログラムのアプリケーションスライスを説明する図である。

【図7】 実行準備状態監視タスク動作のフローチャートを説明する図である。

【図8】 実行準備状態把握処理のフローチャートを説明する図である。

【図9】 低電力モード処理及び高電力モード処理のフローチャートを説明する図である。

【図10】 ビデオ処理タスク及びオーディオ処理タスク動作のフローチャートを説明する図である。

【図11】 タスク内部で行う電力制御アルゴリズムのフローチャートを説明する図である。

【図12】 出力時間管理タスク動作のフローチャートを説明する図である。

【図13】 スリープタスク動作のフローチャートを説明する図である。

【図14】 本発明の情報処理装置の電力制御動作の一実施例を説明する図である。

【図15】 本発明の第2の実施例を説明する概念図である。

【図16】 本発明を施した1プロセッサ構成の情報処理装置の一実施例を説明する図である。

【図17】 本発明を施した2プロセッサ構成の情報処理装置の一実施例を説明する図である。

【図18】 特許文献1の代表図である。

【図19】 RTOSのタスク状態遷移図である。

【図20】 特許文献1に記載の電力制御チェックポイントでの電力制御プログラムの概念図である。

【図21】 割り込まれたタスクの処理遅延を説明する図である。

【図22】 実行状態に遷移した直後のタスクの起動時刻の遅延を説明する図である。

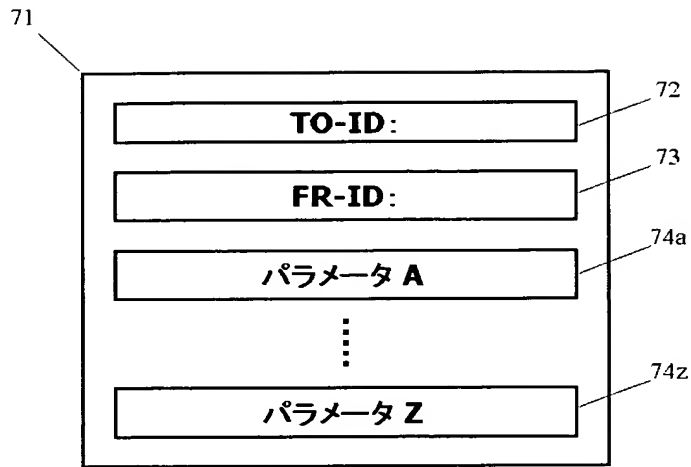
【符号の説明】

【0091】

1・・・アプリケーションプログラム開始点、2・・・初期化处理、3・・・アプリケーションプログラム終了点、4・・・アプリケーションスライス、5・・・繰り返し判別処理、6・・・電力制御チェックポイント、7・・・アプリケーションスライス処理経過管理部、8・・・画像フォーマット、9・・・マクロブロック、11・・・出力時間、12・・・タスク起動時刻、13・・・サブ出力時間、14・・・WCET、15・・・積算実行時間、16・・・処理終了時点、21・・・プロセッサ、22・・・電力制御回路または可変DC/DC Converter、23・・・RTOS、24、64・・・タスク、25・・・ソフトウェア、31・・・実行状態、32・・・待機状態、33・・・実行準備状態、34・・・初期状態、35・・・ディスパッチ、36・・・プリエンプト、37・・・実行時間、38・・・待機時間、39・・・遅延起動時刻、40・・・遅延WCET、41・・・仮想出力時間、42・・・仮想WCET、43・・・仮想サブ出力時間、51・・・CPG、52・・・TMU、53・・・バス、54・・・メモリ、55・・・I/O、56・・・実行準備状態監視プログラムモジュール、57・・・電力/エネルギー制御プログラムモジュール、58・・・アプリケーション/ミドルウェアプログラムモジュール、59・・・バッテリー、60・・・電源系統、61・・・制御信号、62・・・ポート、65・・・System Call、66・・・RWI、67・・・PECI、68・・・API、71・・・メッセージ、72・・・送信先宛先、73・・・送信元宛先、74・・・送信内容、81・・・初期状態、82・・・実行準備状態監視変数部、83・・・メッセージ受信待ち判別処理、84・・・実行準備状態把握処理、91・・・実行準備状態監視開始点、92・・・メッセージ判別処理、93・・・実行準備状態加算処理、94・・・実行準備状態減算処理、95・・・実行準備状態監視変数判別処理、96・・・低電力モード処理、97・・・高電力モード処理、101・・・低電力モード処理開始点、102・・・低動作周波数変更処理、103・・・低電源電圧変更処理、104・・・高電力モード処理開始点、105・・・高電源電圧変更処理、106・・・高動作周波数変更処理、111・・・ビデオ処理初期化開始点、112・・・ビデオ復号化処理、113・・・待機状態遷移処理、121・・・タスク内電力制御開始点、122・・・実行準備状態監視変数判別処理、123・・・WCET比較処理、124・・・仮想WCET比較処理、125・・・電力制御モード部、126・・・WCET格納部、127・・・仮想WCET格納部、128・・・電力制御モード判別処理、131・・・出力時間管理開始点、132・・・出力時間管理処理、141・・・スリーププログラム開始点、142・・・スリープ処理、151・・・アンテナ、152・・・RF、153・・・マイクロフォン、154・・・スピーカー、155・・・CMOSまたはCCDカメラ、156・・・液晶または有機ELディスプレイ、157・・・動的メモリエリア。

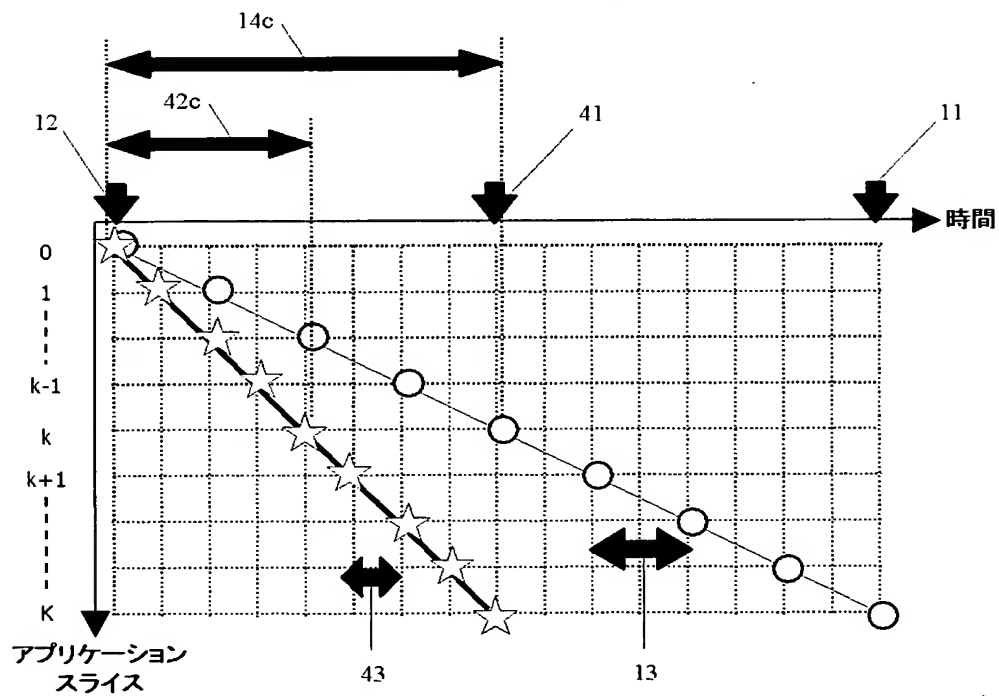
【図 3】

図 3

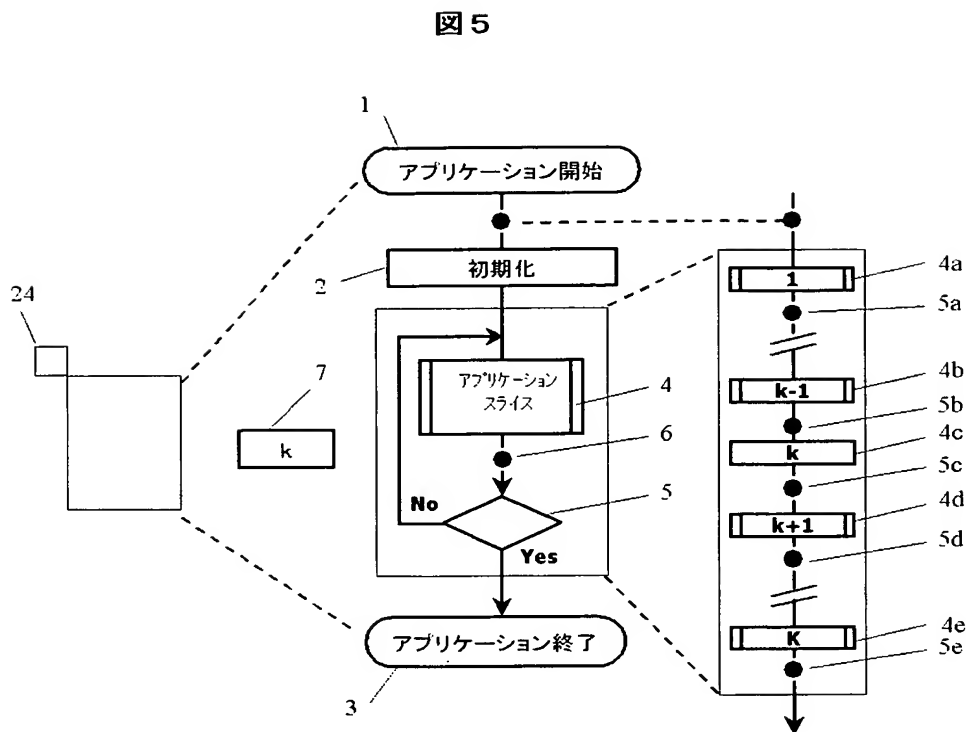


【図 4】

図 4

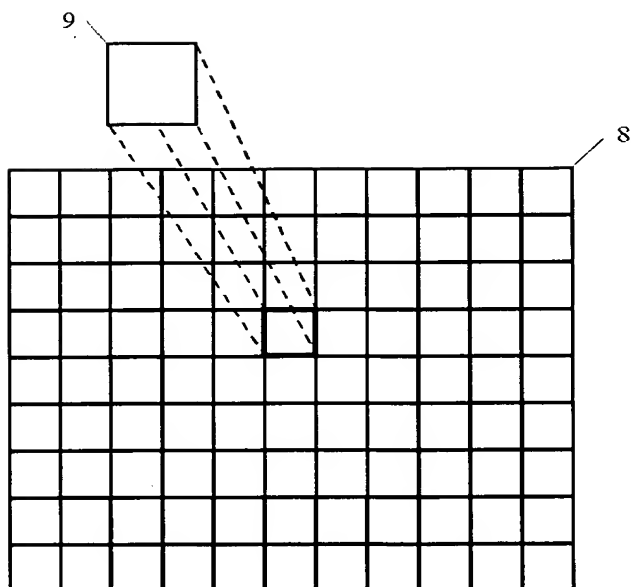


【図 5】



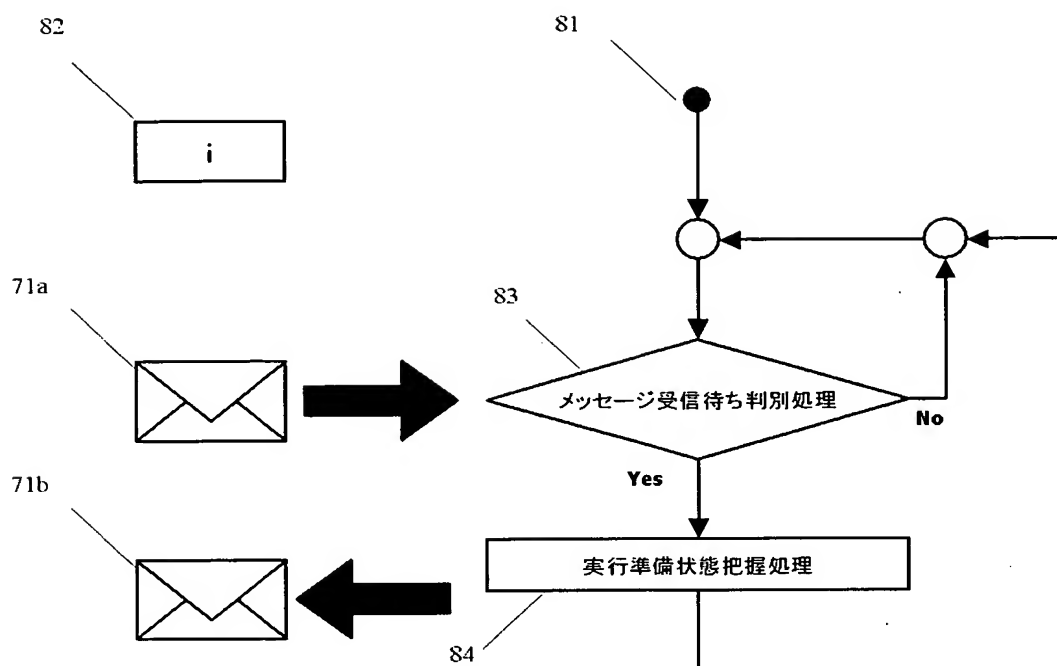
【図 6】

図 6



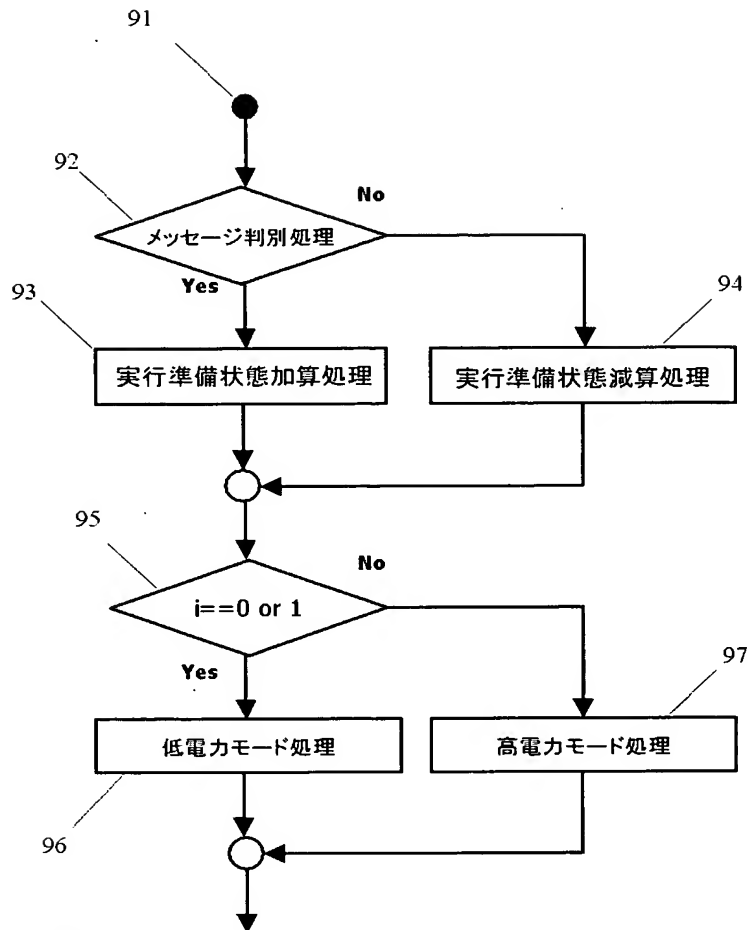
【図 7】

図 7



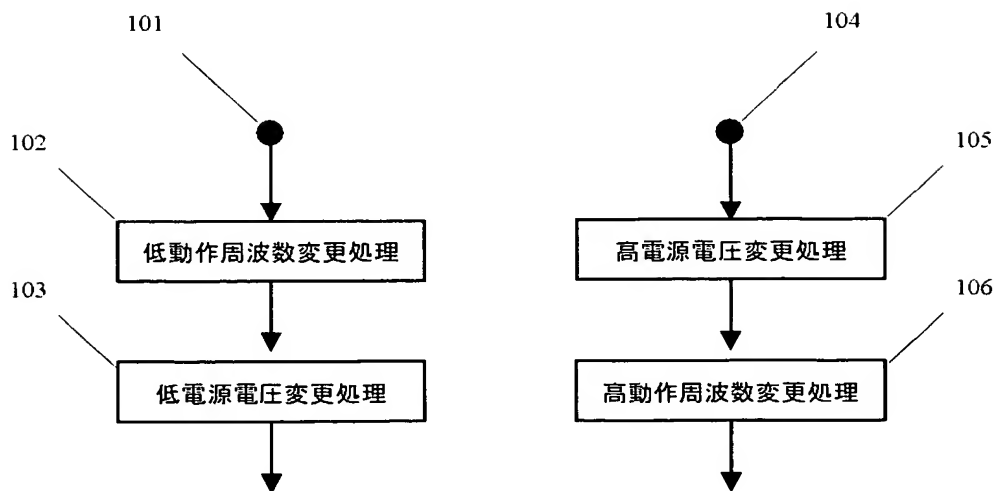
【図 8】

図 8



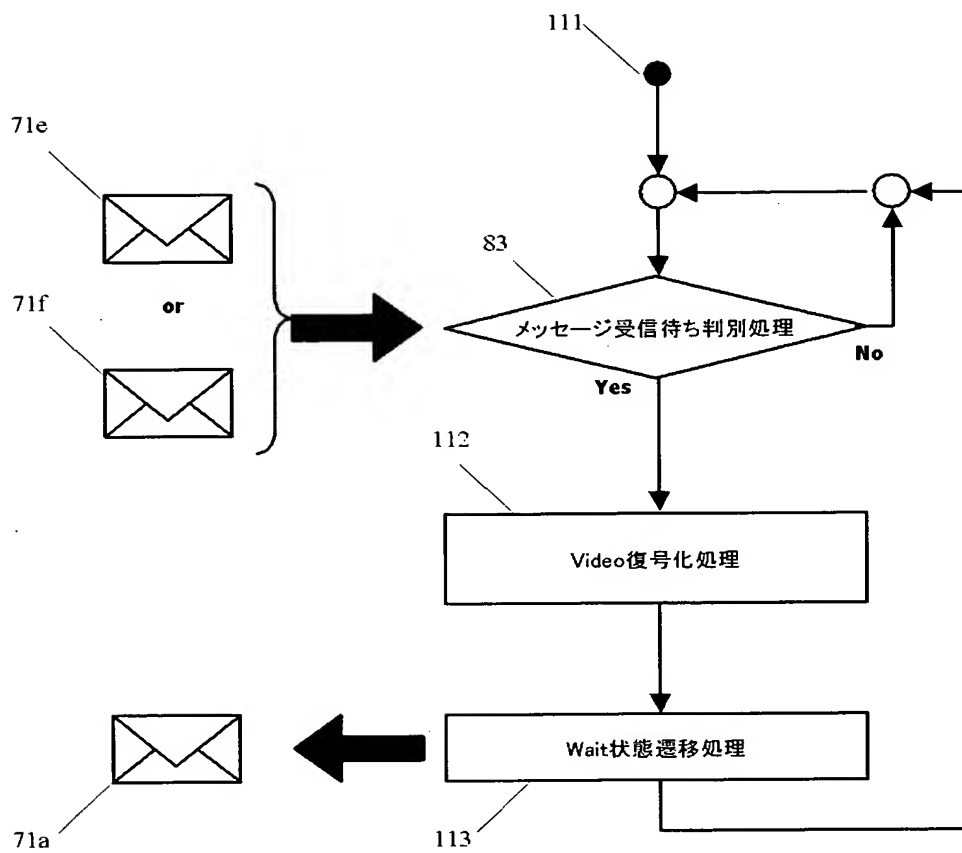
【図 9】

図 9



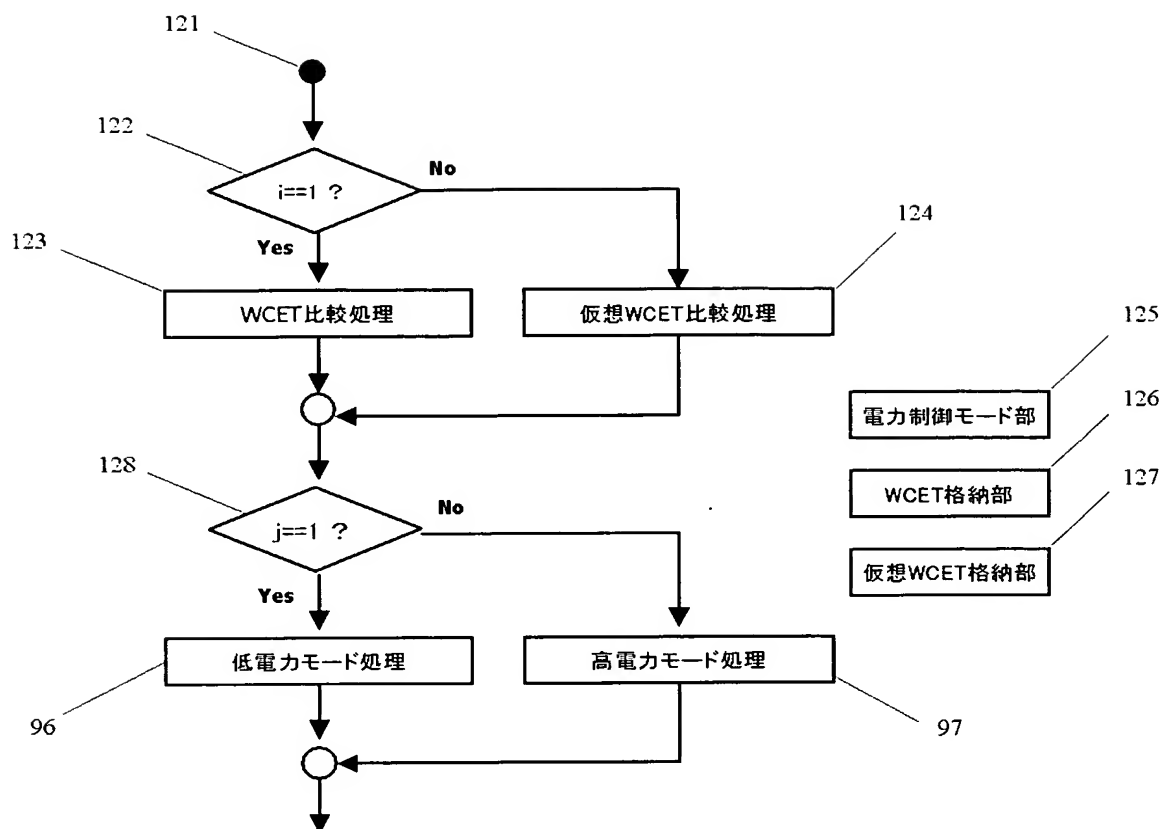
【図 10】

図 10



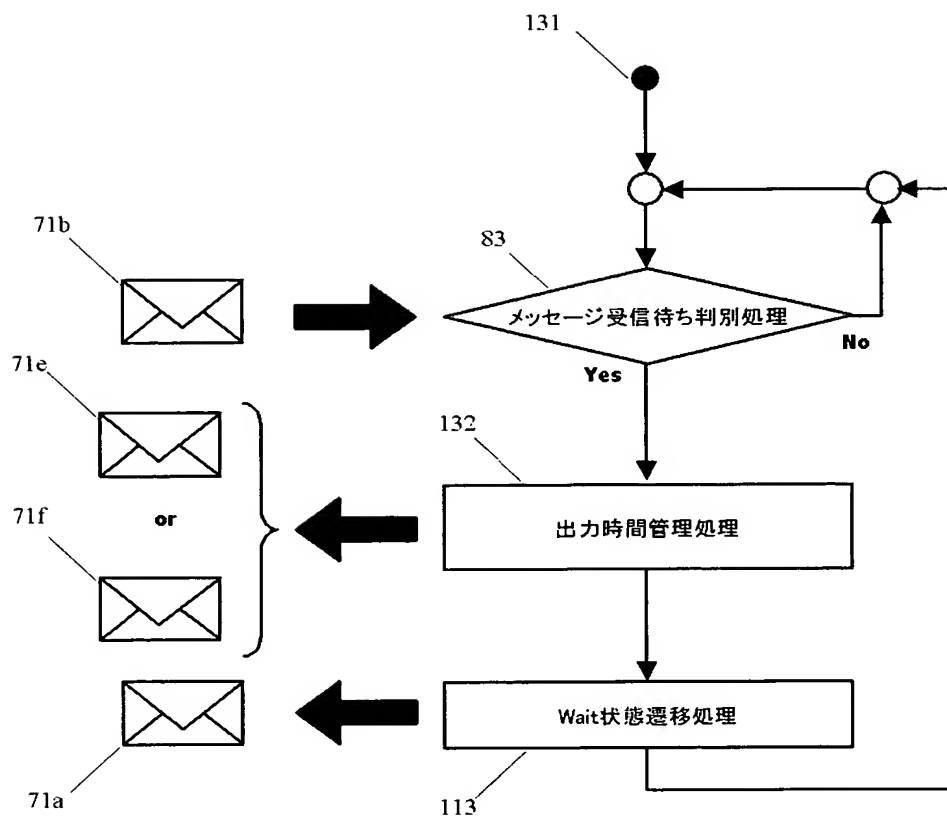
【図 11】

図 11



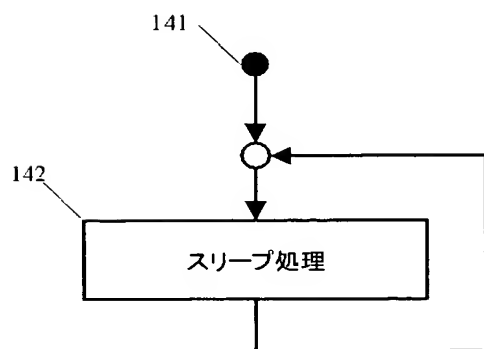
【図 1 2】

図 1 2



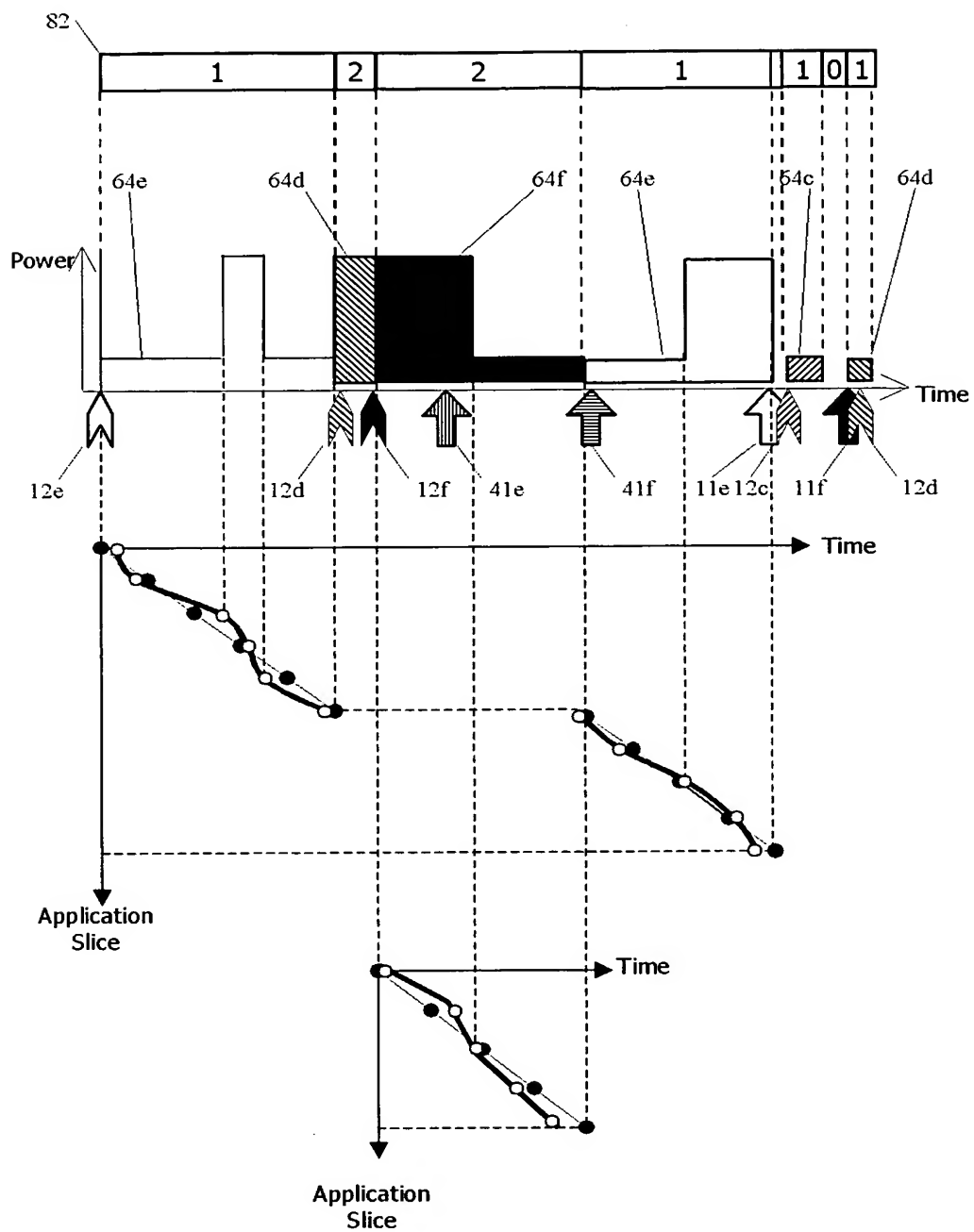
【図 1 3】

図 1 3



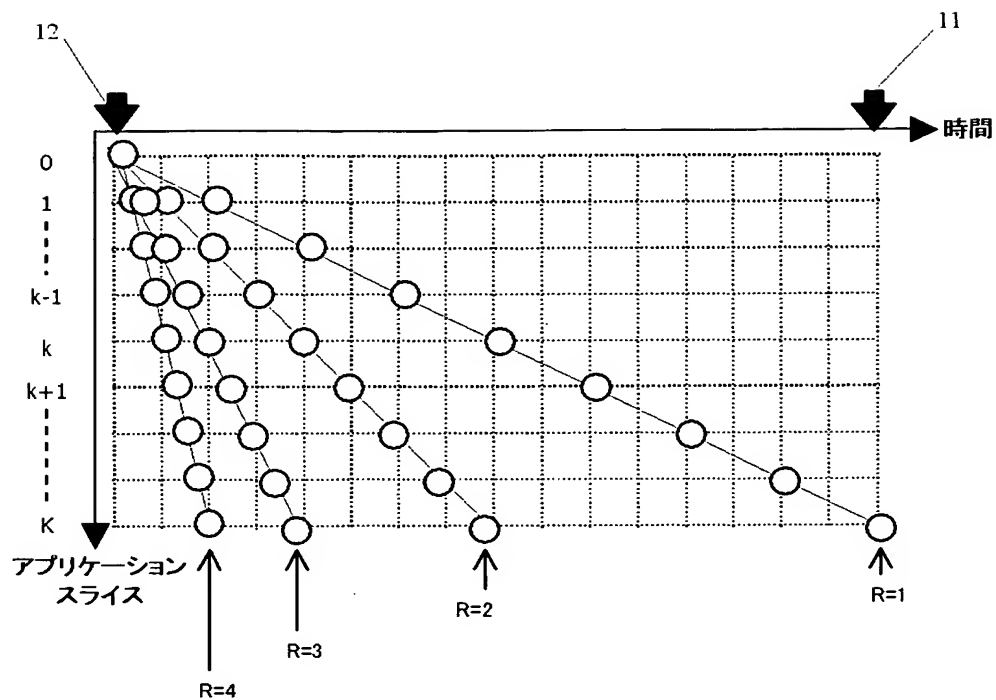
【図 14】

図 14



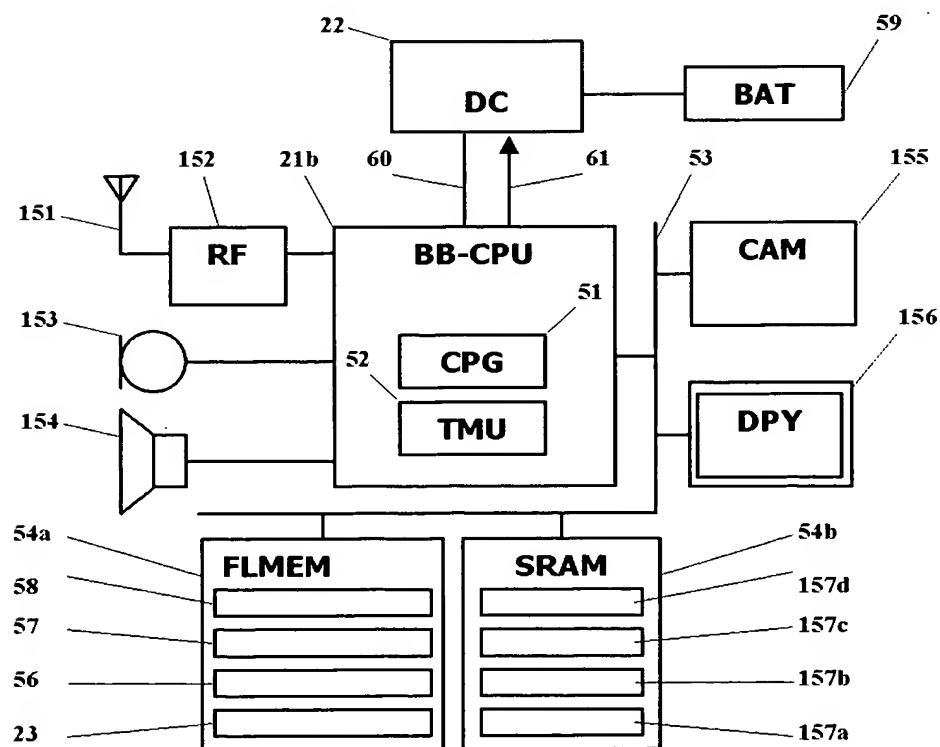
【図 15】

図 15



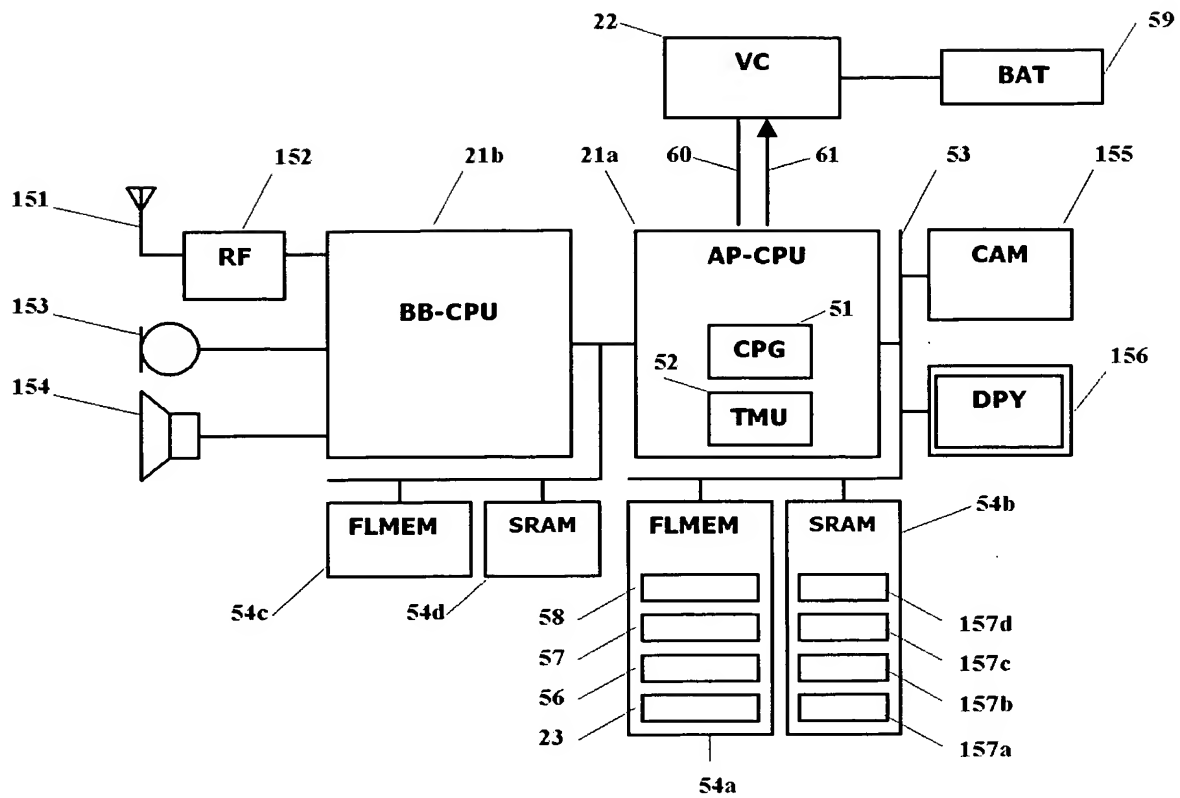
【図 16】

図 16



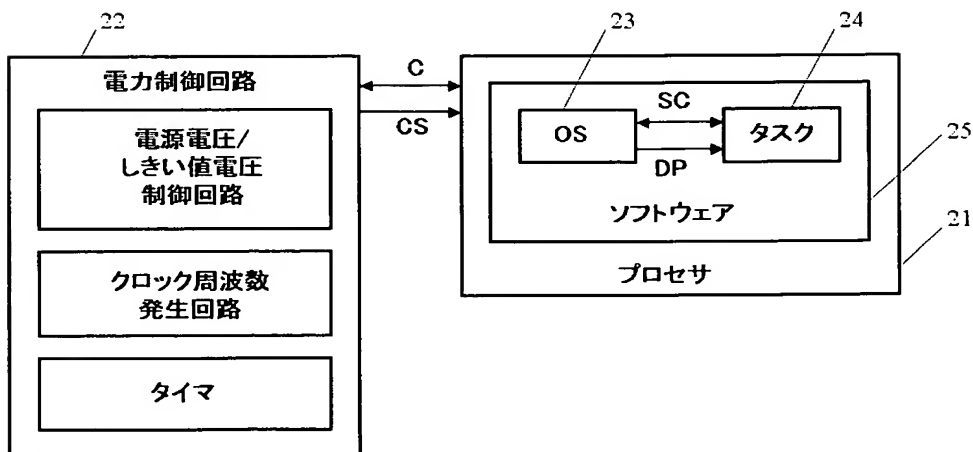
【図 17】

図 17



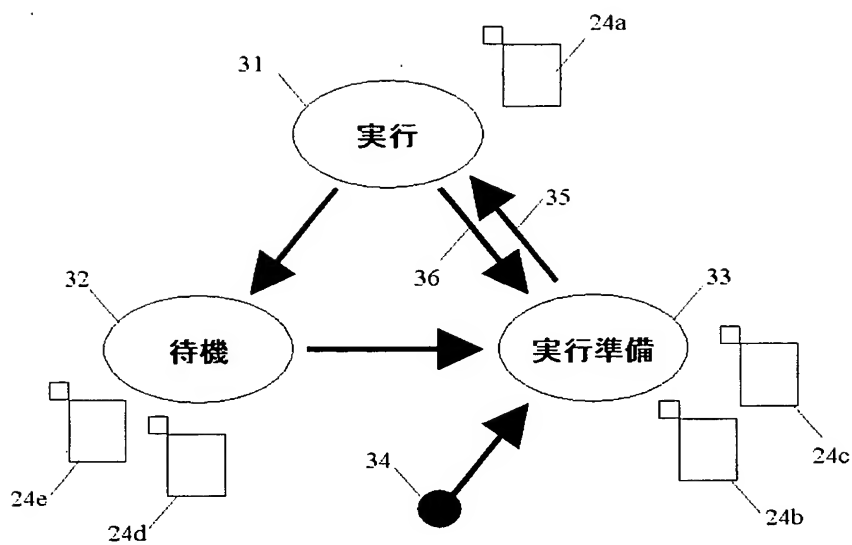
【図 18】

図 18



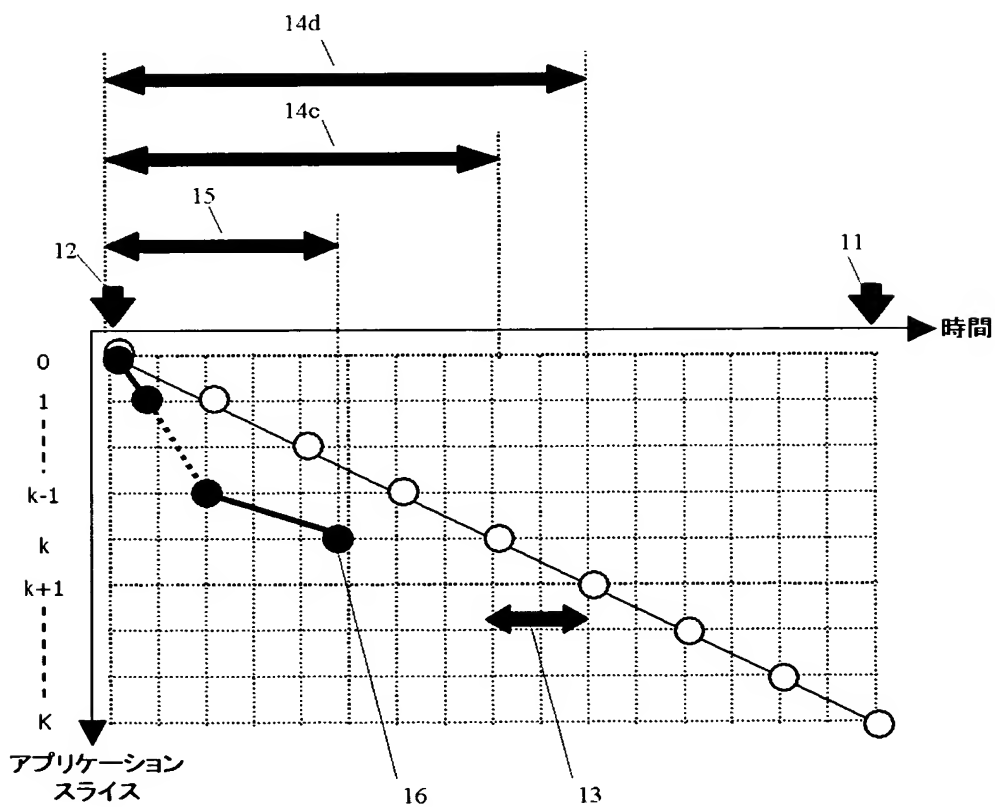
【図 19】

図 19



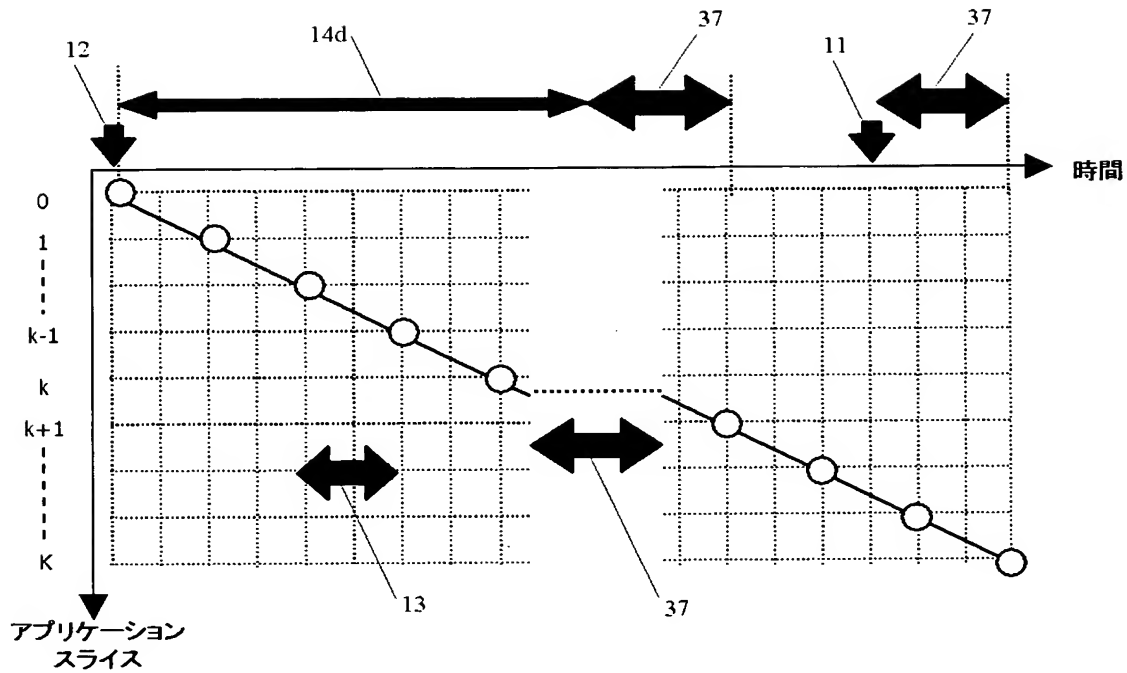
【図 20】

図 20



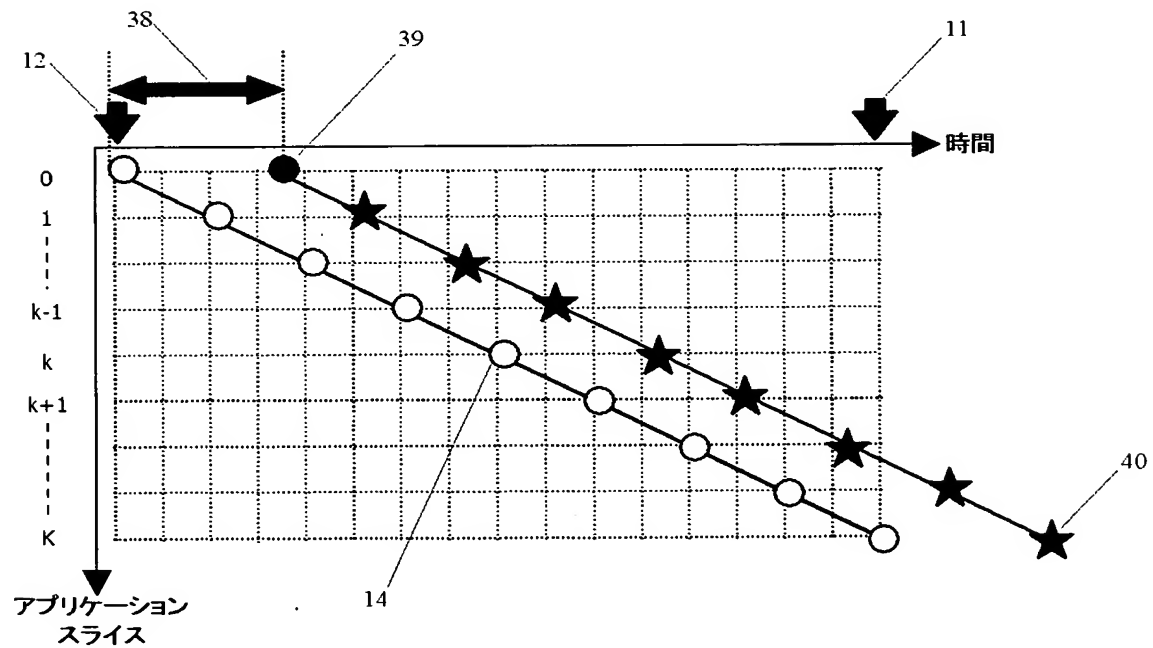
【図 2 1】

図 2 1



【図 2 2】

図 2 2



【書類名】 要約書

【要約】

【課題】 割込/例外処理が発生してもよりRobustな電力制御を行うことにより、低消費電力を実現する情報処理装置を提供する。

【解決手段】 RTOSまたは実行準備状態監視タスクによって実行準備状態のタスクの数を監視した結果、実行準備状態のタスクが全く無い場合は、RTOSまたは実行準備状態監視タスクで電力を下げる制御を施し、そして実行中のタスクは予め設定した各アプリケーションスライスのWCETで電力制御を行い、反対に実行準備状態のタスクがある場合は、電力を上げる制御を施し、そして実行中のタスクは各アプリケーションスライスのWCETよりも時刻の早い仮想WCETで電力制御を行い、更に実行準備状態のタスクが無く且つ実行中のタスクも存在しない場合は、RTOSまたは実行準備状態監視タスクとスリープタスクが電力を下げる。

【効果】 情報処理装置の低消費電力化を向上させること達成することが出来る。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 4 1 8 9 0 8
受付番号	5 0 3 0 2 0 7 3 0 3 2
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 1 2 月 1 8 日

< 認定情報・付加情報 >

【提出日】	平成 15 年 12 月 17 日
-------	-------------------

特願 2 0 0 3 - 4 1 8 9 0 8

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 1 0 8]

1. 変更年月日	1 9 9 0 年 8 月 3 1 日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台 4 丁目 6 番地
氏 名	株式会社日立製作所